

BEST AVAILABLE COPY

FIG. 1

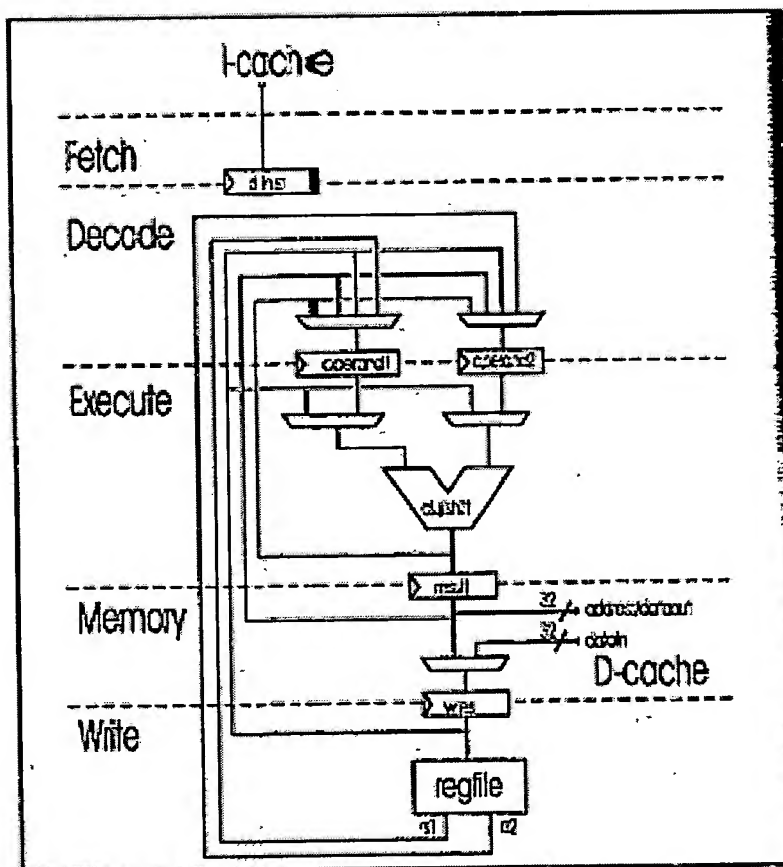


FIG. 2

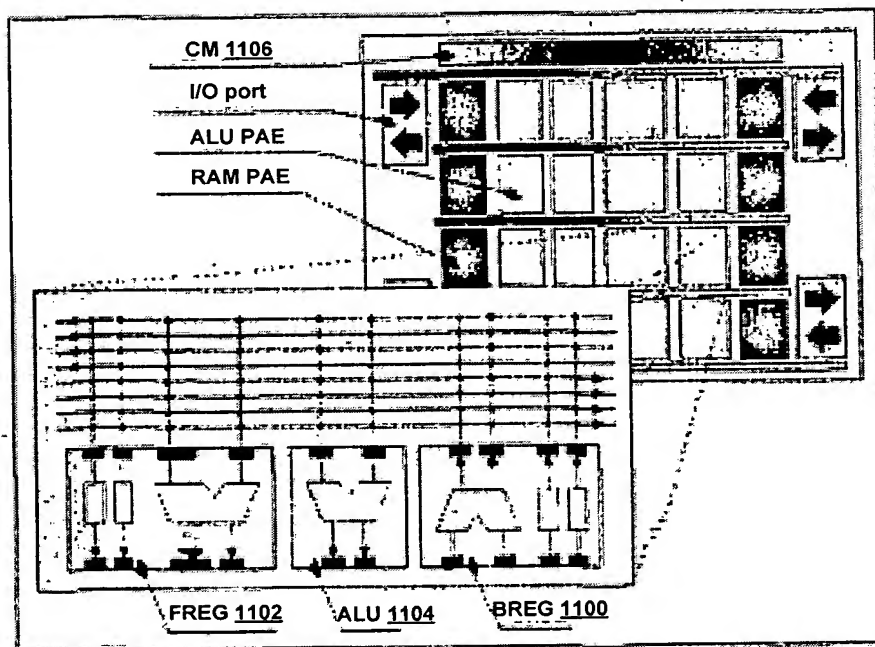


FIG. 3

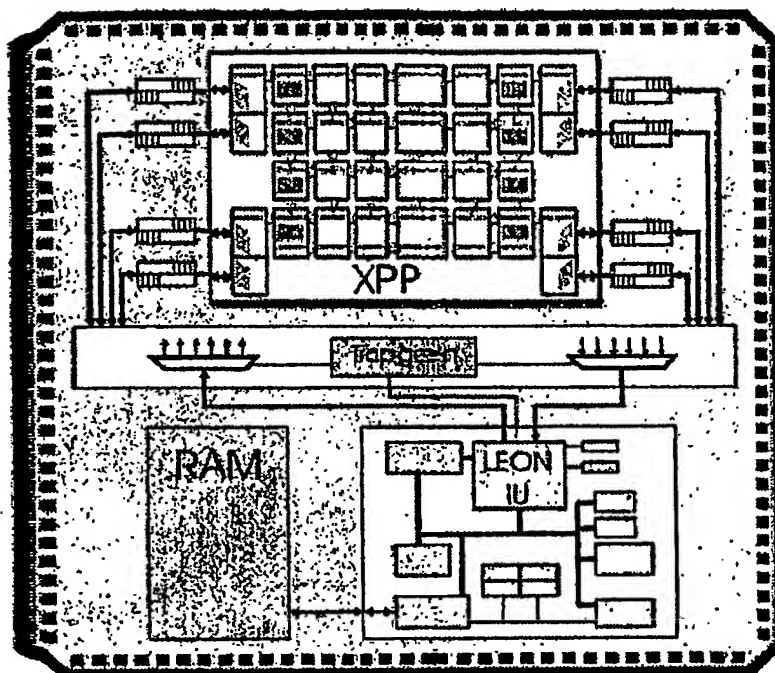


FIG. 4

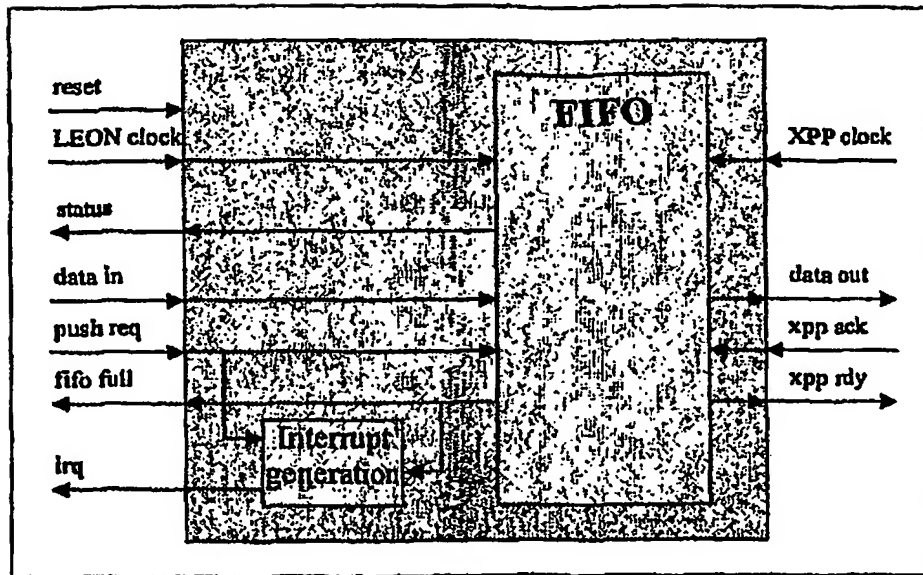


FIG. 5

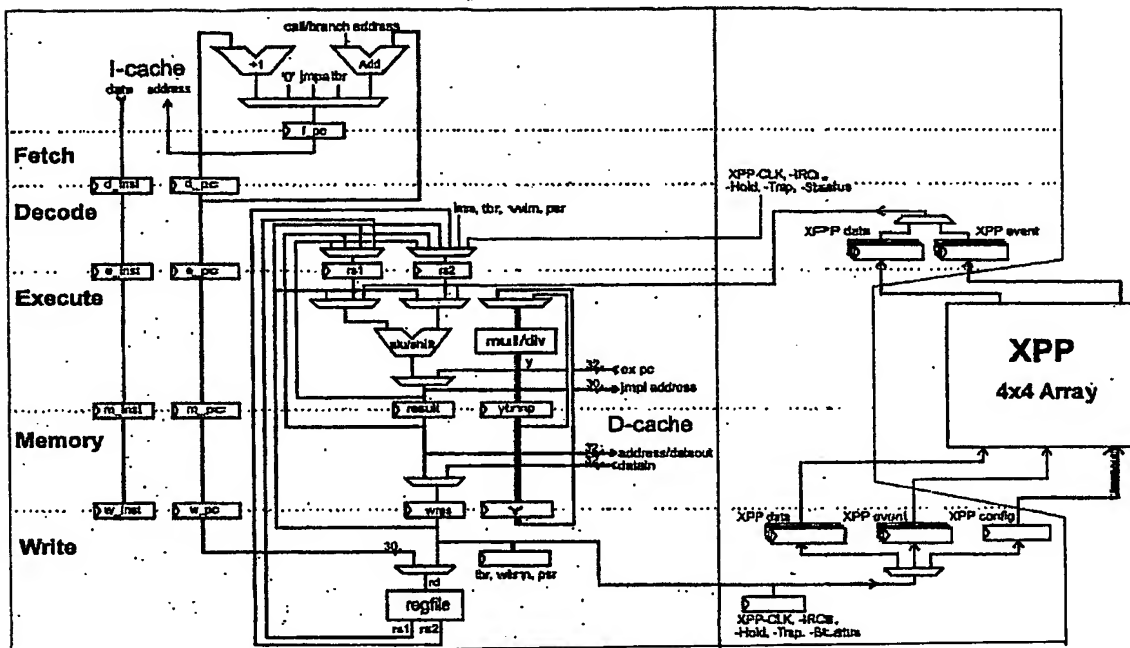


FIG. 6

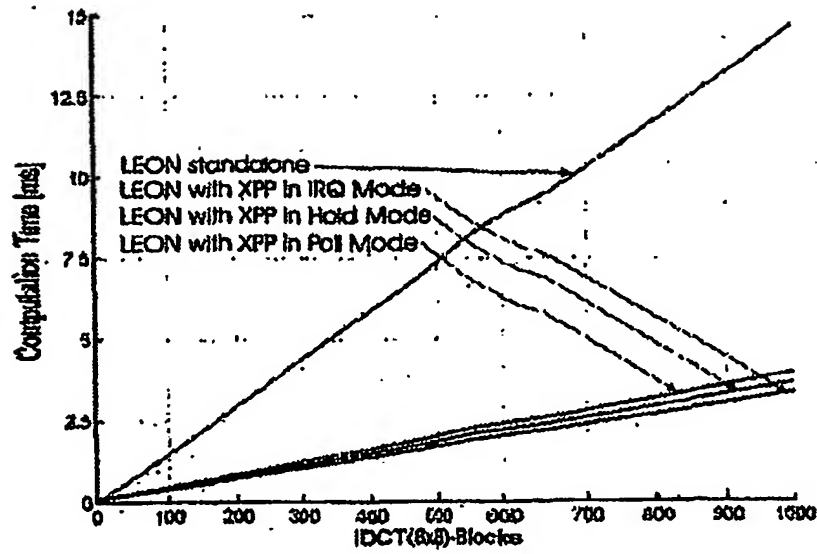


FIG. 7

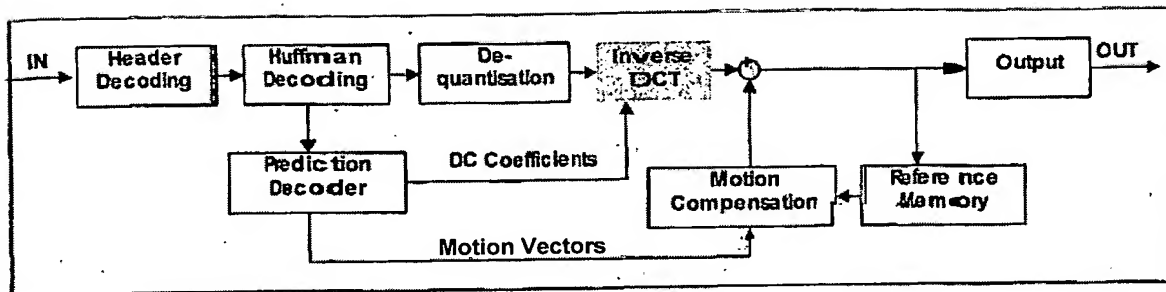


FIG. 8

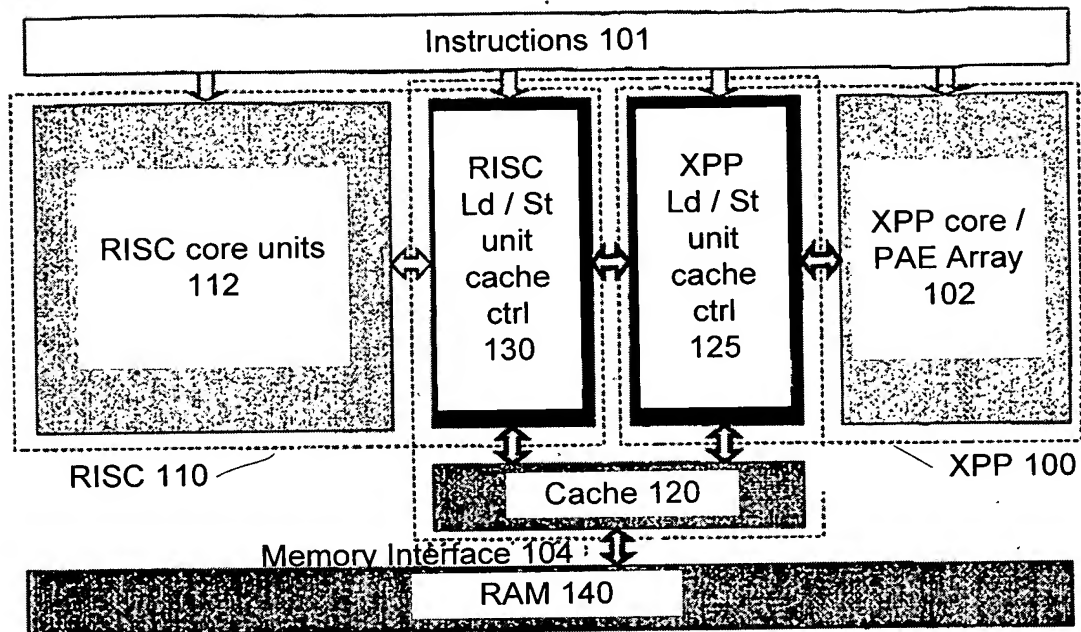


FIG. 9

```
XPPP reloadConfig(_XppCfg_foo );
For (int i=0; i < 1000; ++i) {
  XPPP Preload ( 2, &a[i*30], 30 );
  XPPP Preload ( 0, &b[i*200], 200 );
  XPP PreloadClean( 5, &c[i*10], 10 );
  XPP Execute();
}
```

/*
Other RISC computations . . .
In the meanwhile the burst preloads
and the previous configuration are
running;
The new configuration is executed
as soon as the preloads and the
previous configuration are finished.
New burst preloads can be issued
according to the FIFO length.

Note: in all places where constants
are used, the value should actually
come from a register.

*/
}

Legend:

per thread state resource
volatile (non-state resource write back if dirty
volatile read only resource

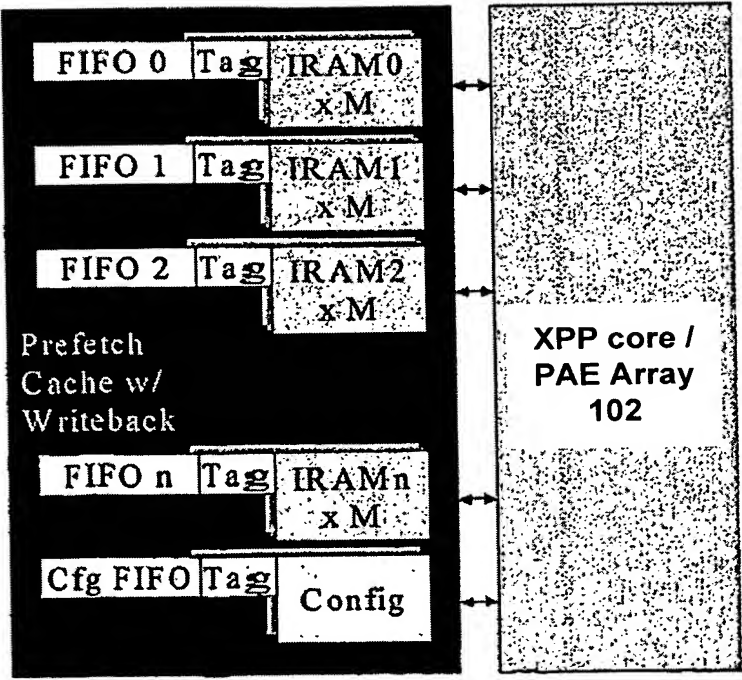
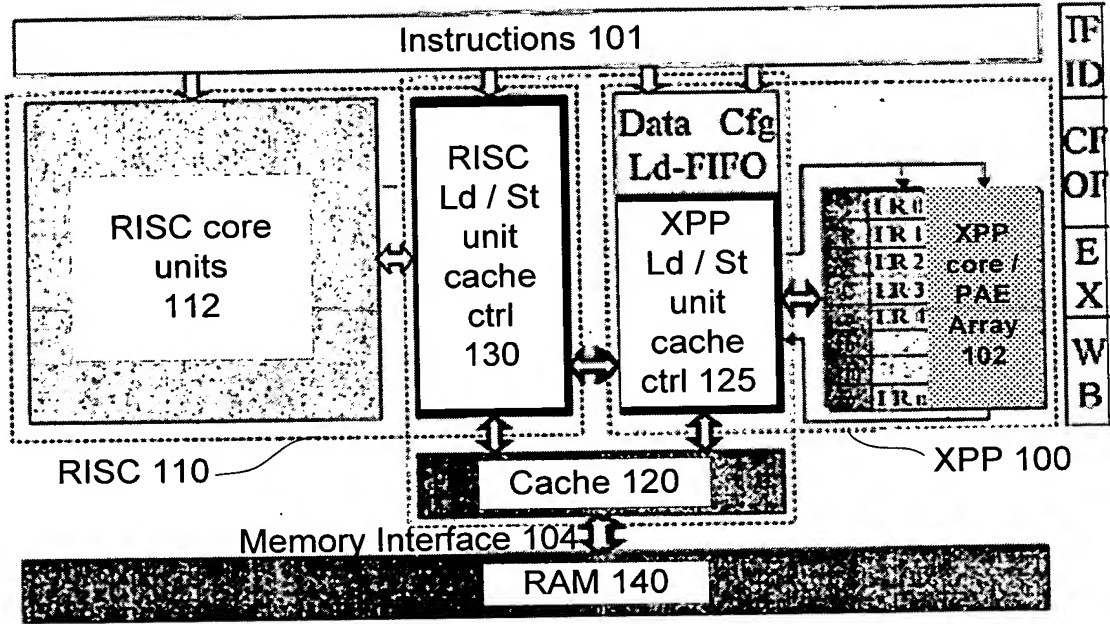


FIG. 10



IF
ID
CF
OF
EX
WB

FIG. 11

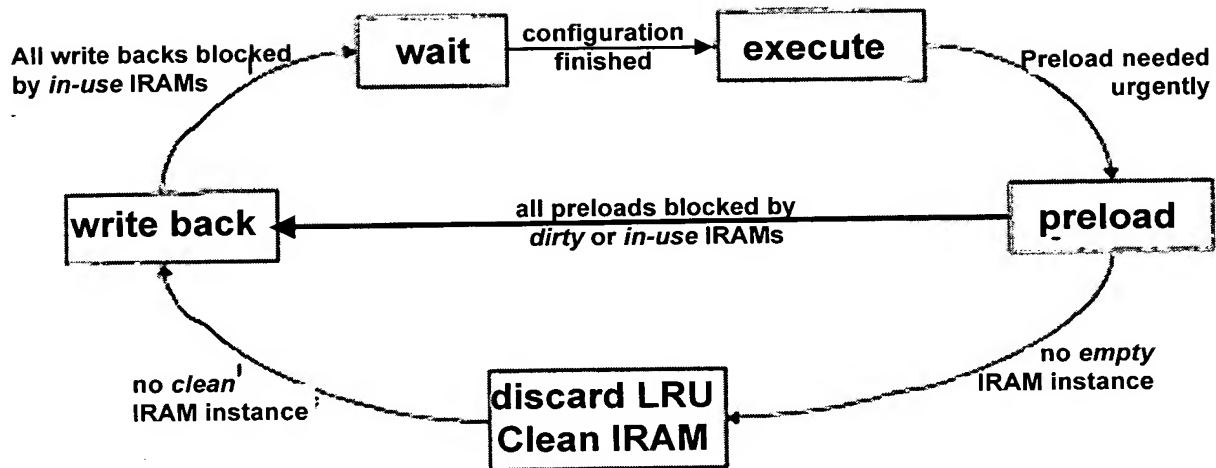


FIG. 12

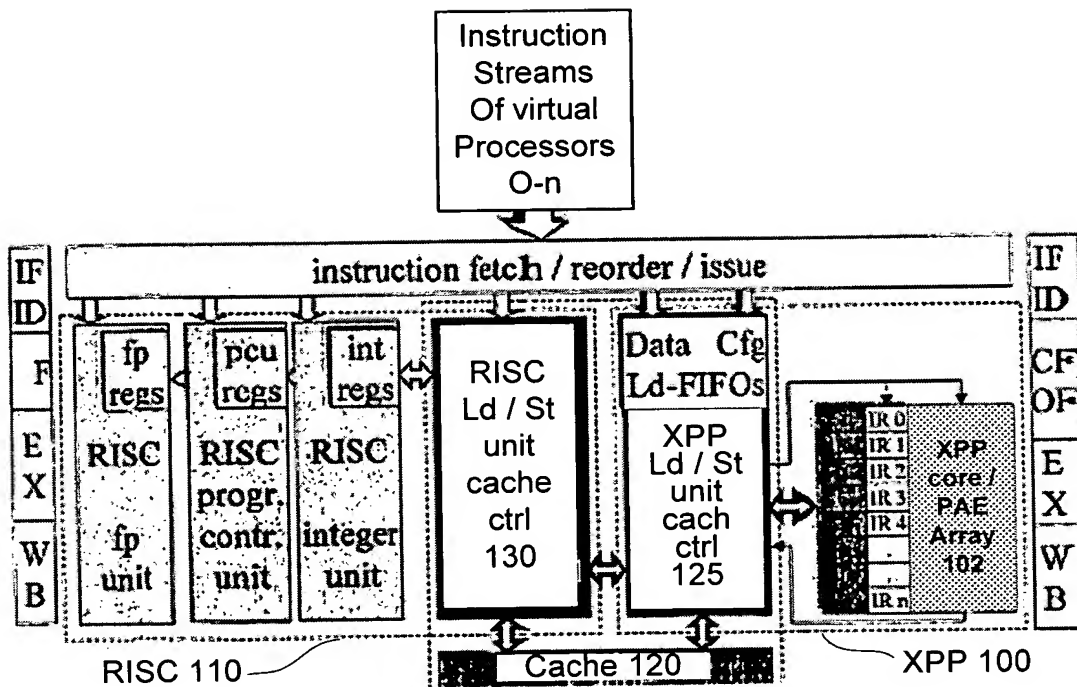


FIG. 13

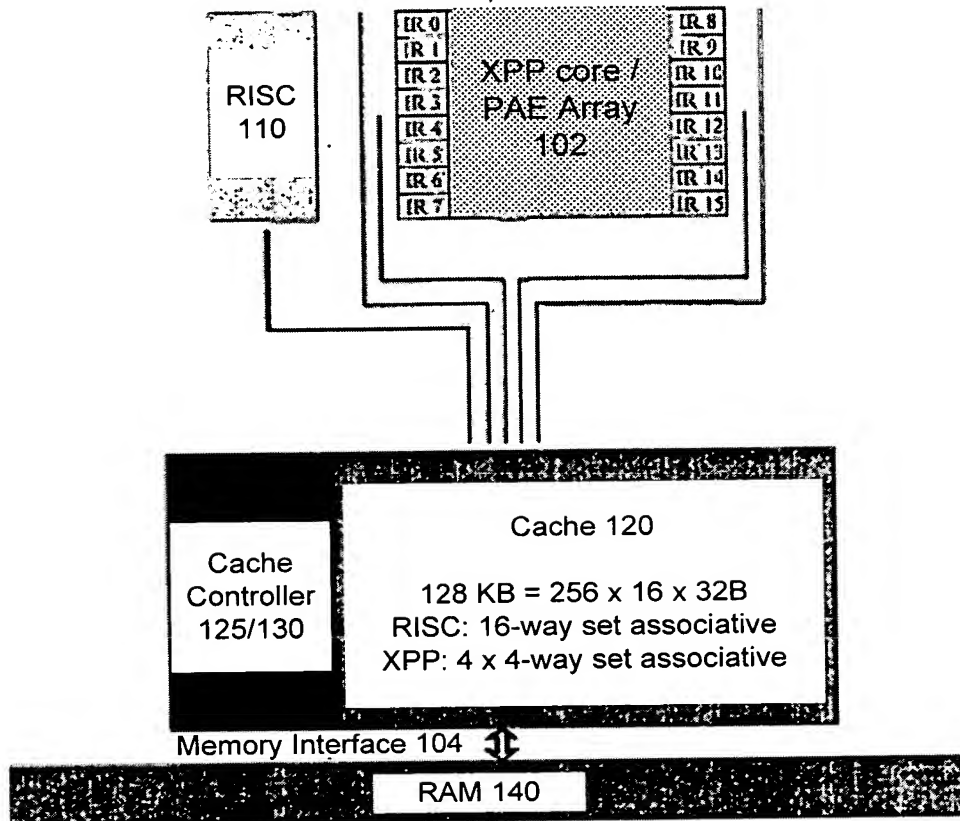


FIG. 14

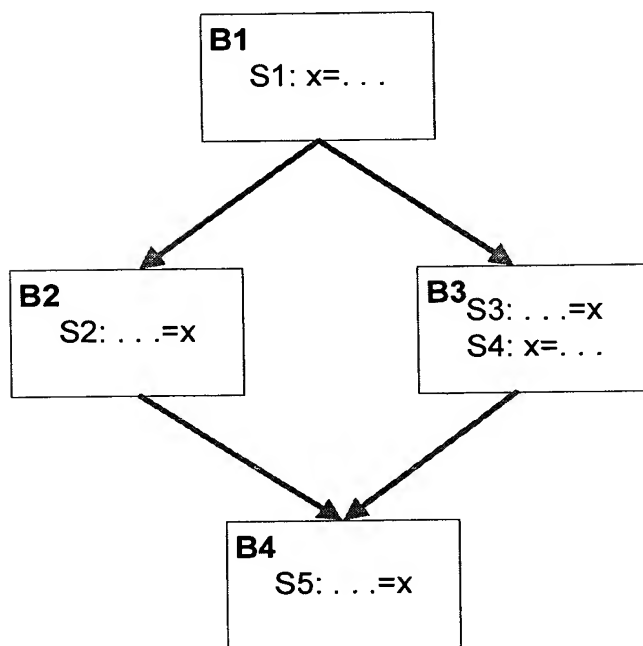


FIG. 15

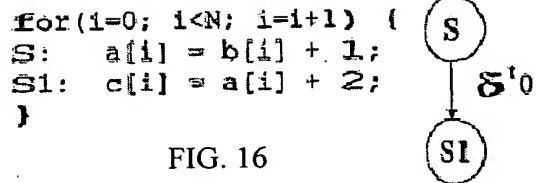


FIG. 16

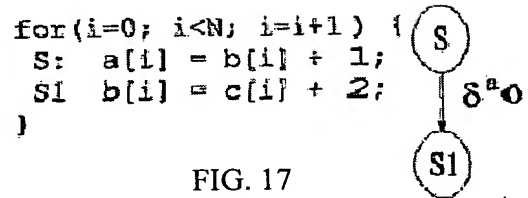


FIG. 17

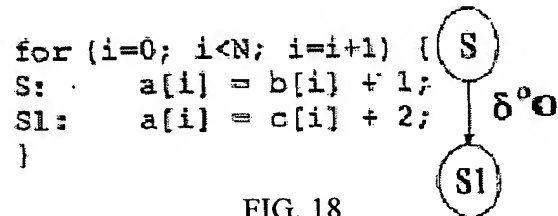


FIG. 18

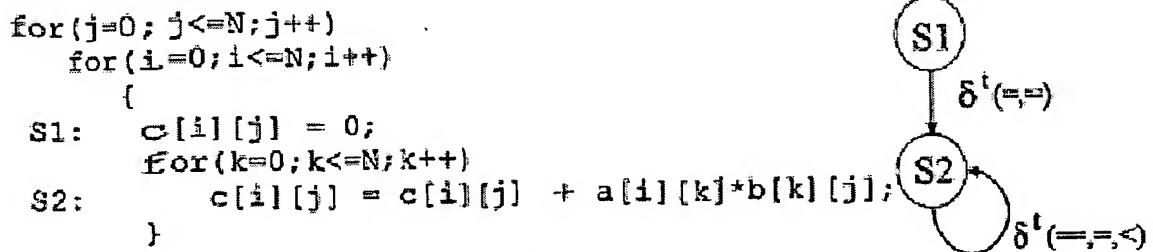


FIG. 19

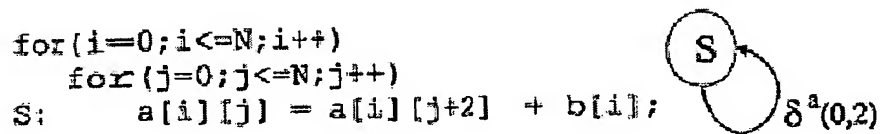


FIG. 20

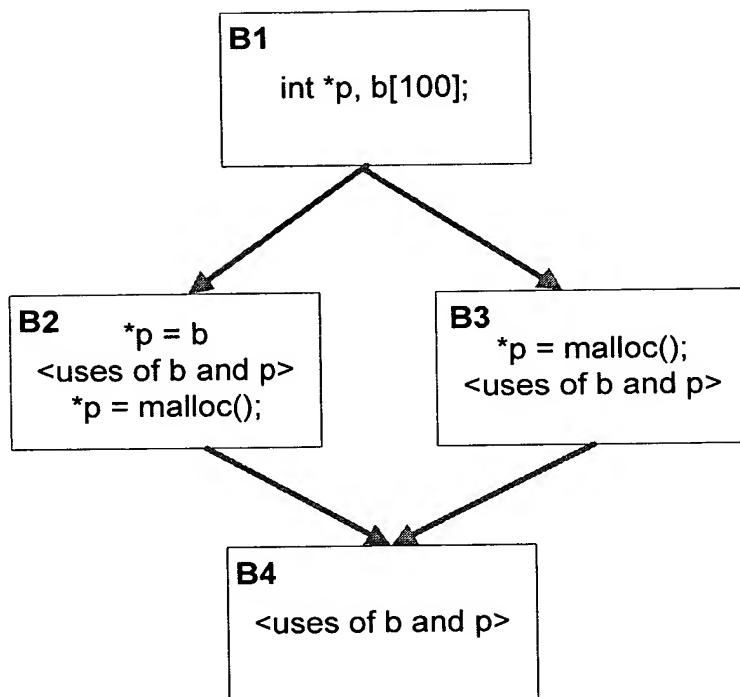


FIG. 21

Application No. 10/523,764
Preliminary Amendment
Replacement Sheet 18

0	4	8	12	0	4	8	12
16	20	24	28	16	20	24	28
32	36	40	44	32	36	40	44
48	52	56	60	48	52	56	60

FIG. 22

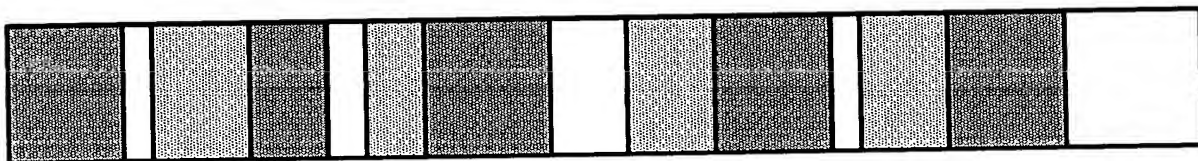


FIG. 23

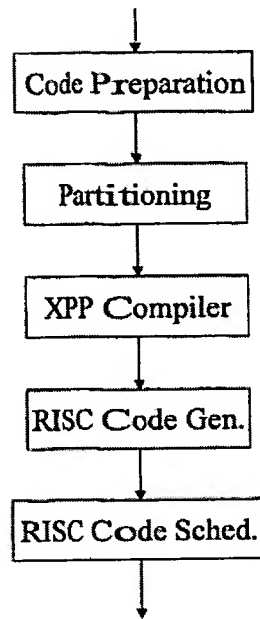


FIG. 24

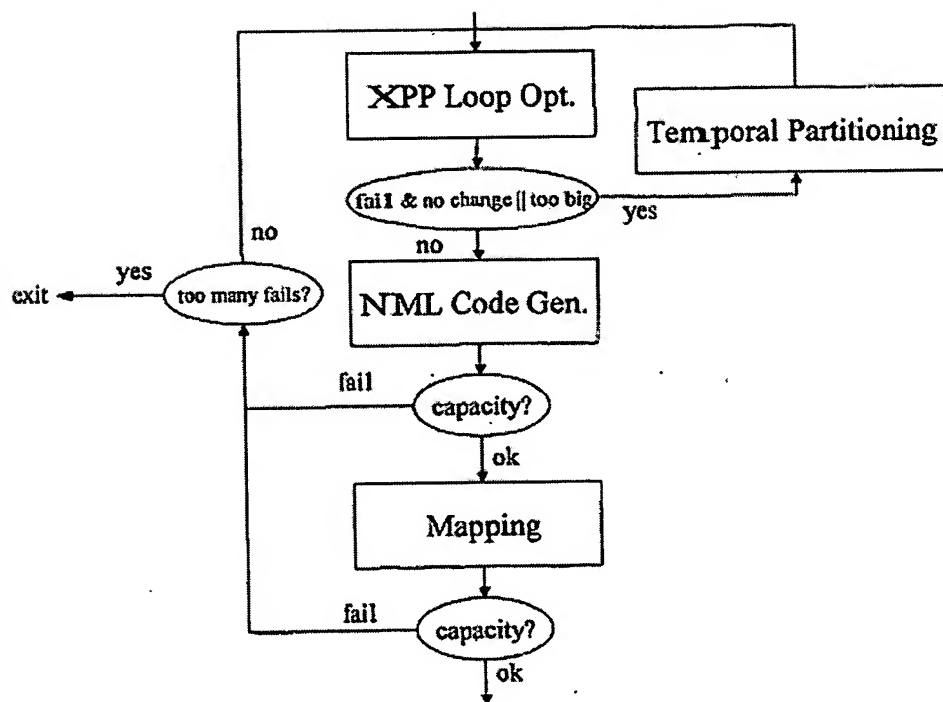


FIG. 25

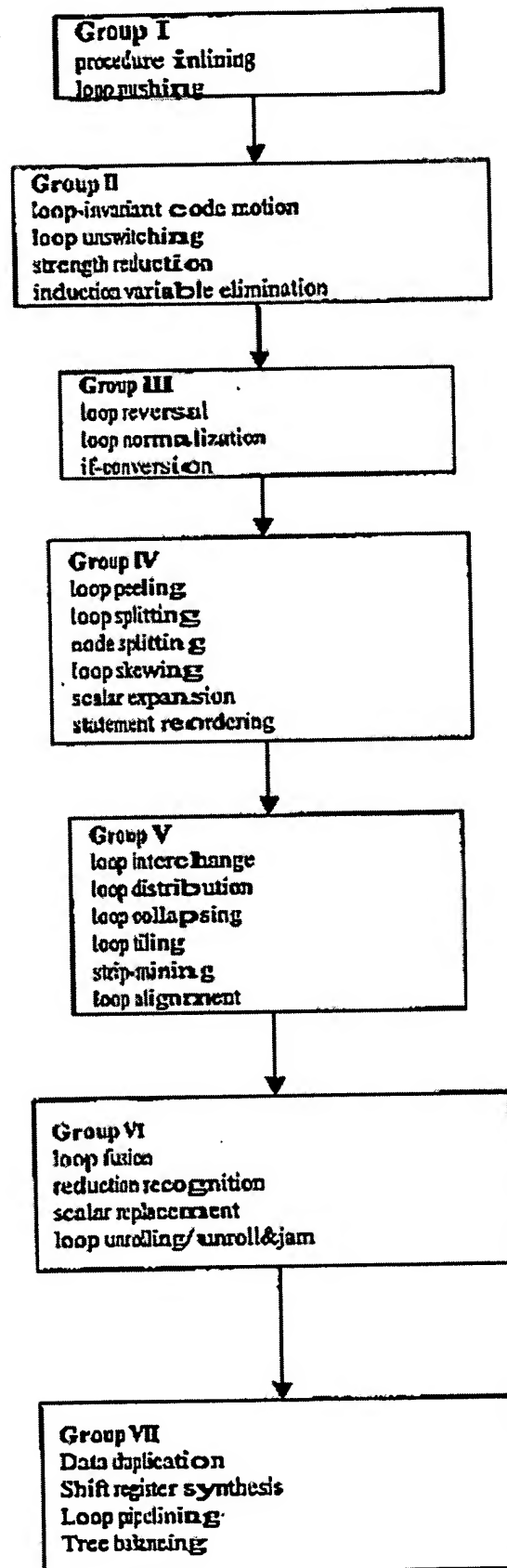


FIG. 26

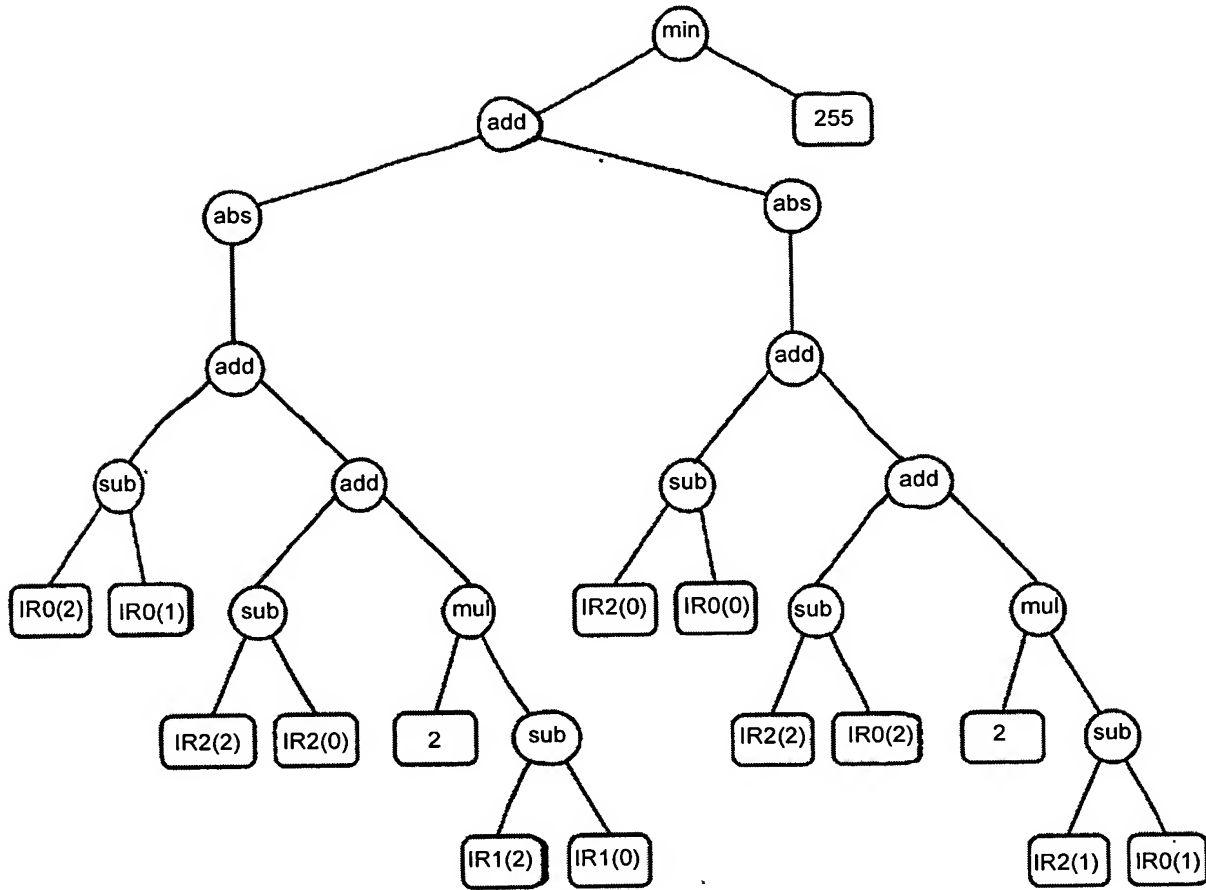


FIG. 27

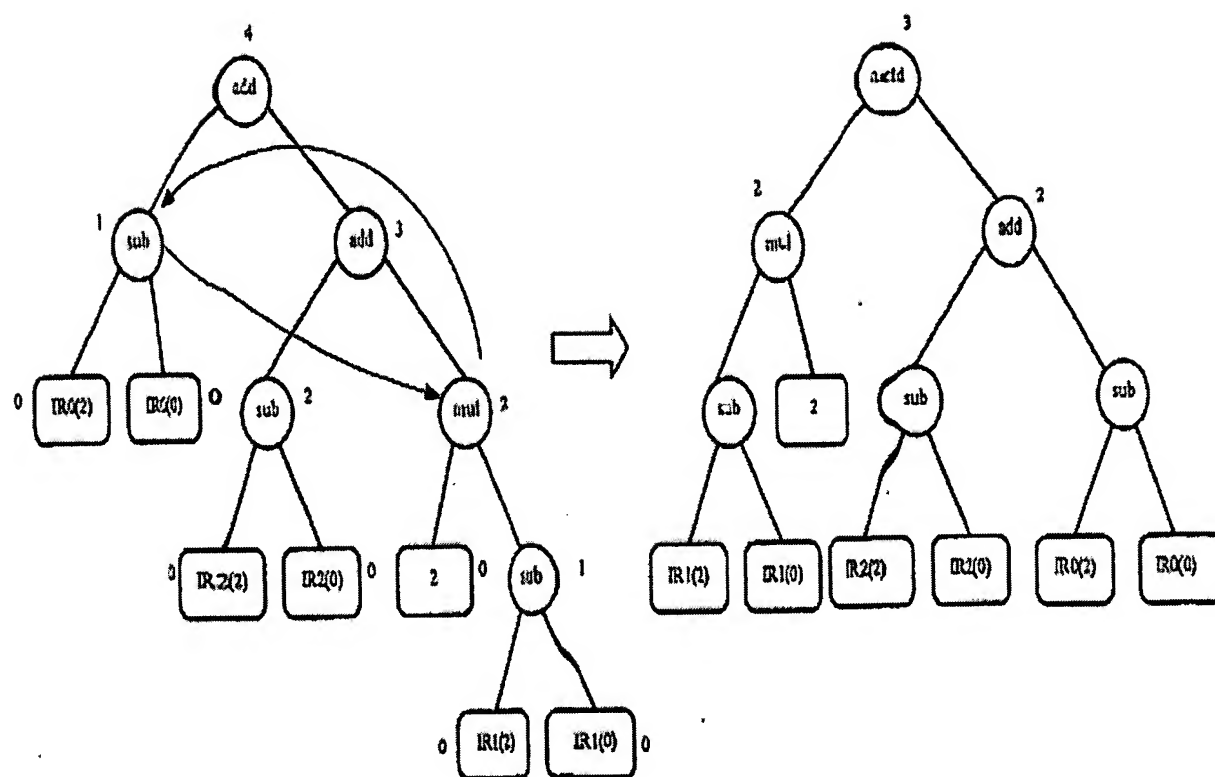


FIG. 28

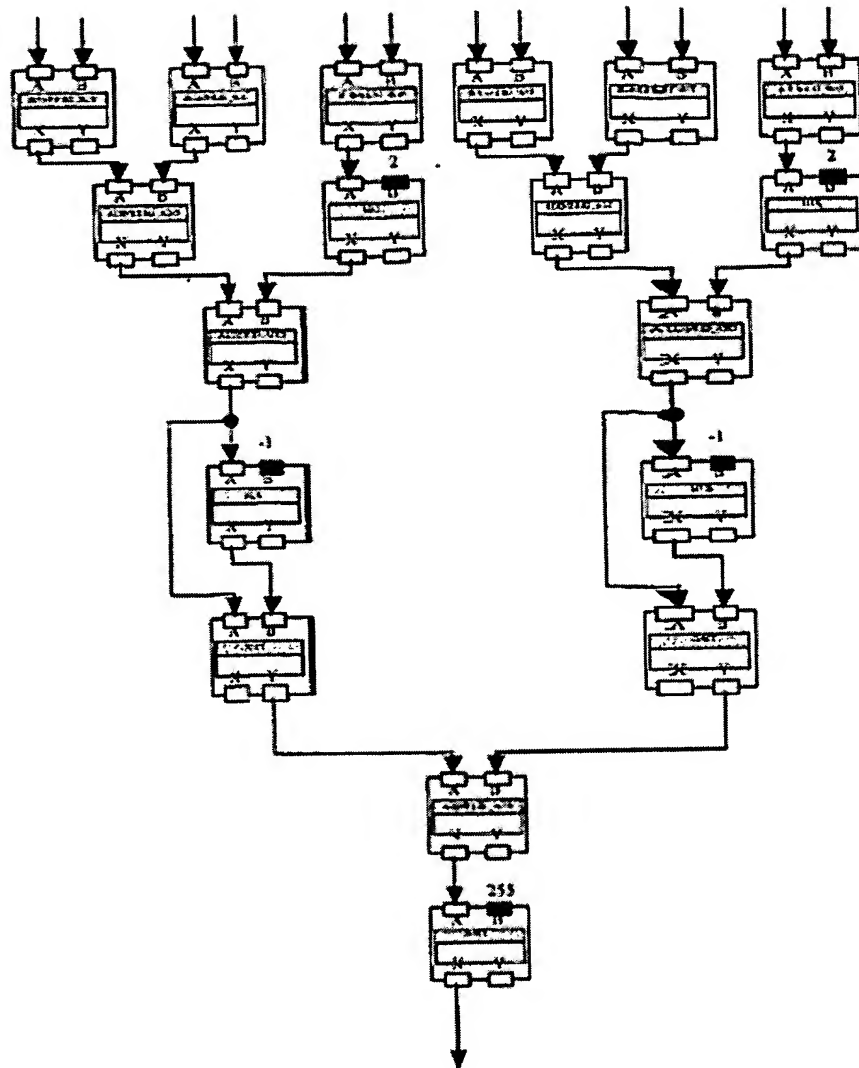


FIG. 29

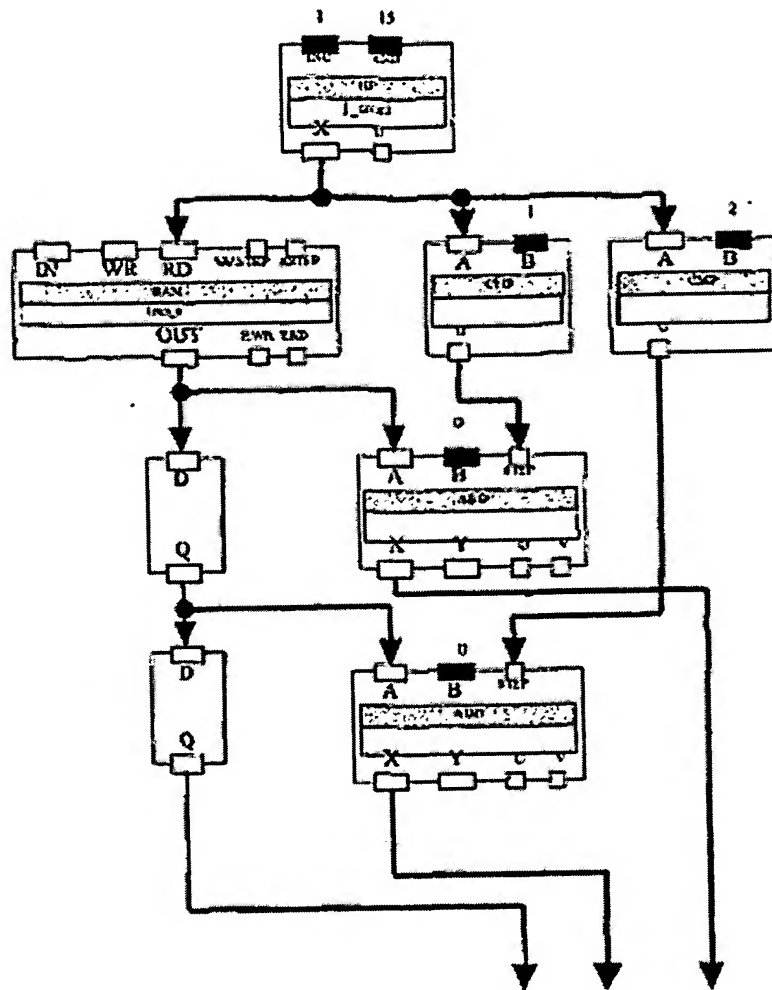


FIG. 30

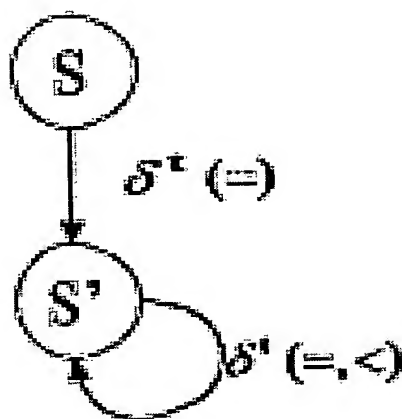


FIG. 31

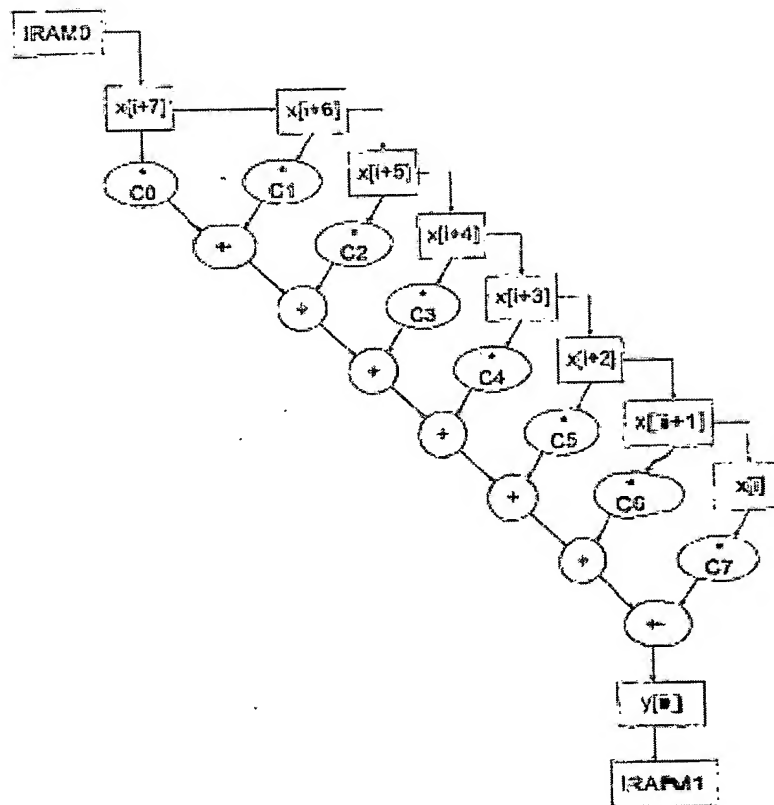


FIG. 32

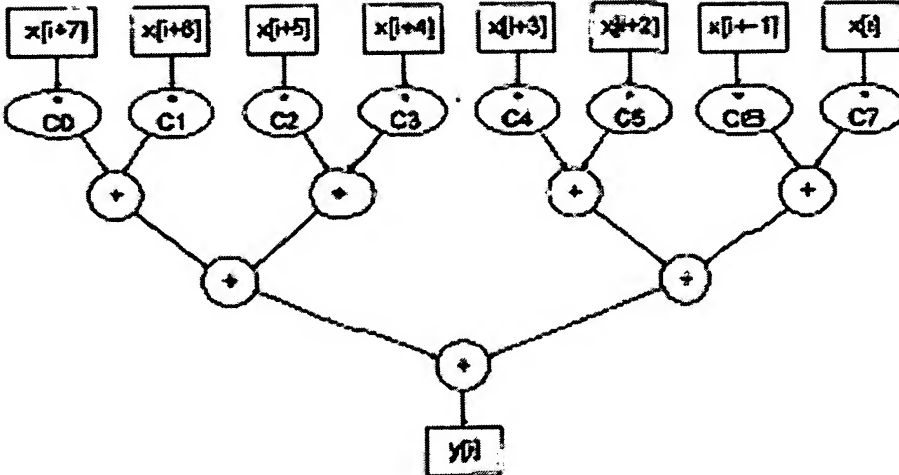


FIG. 33

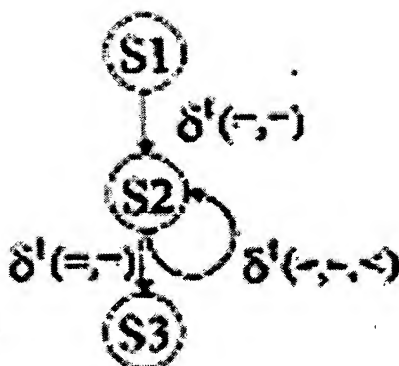


FIG. 34

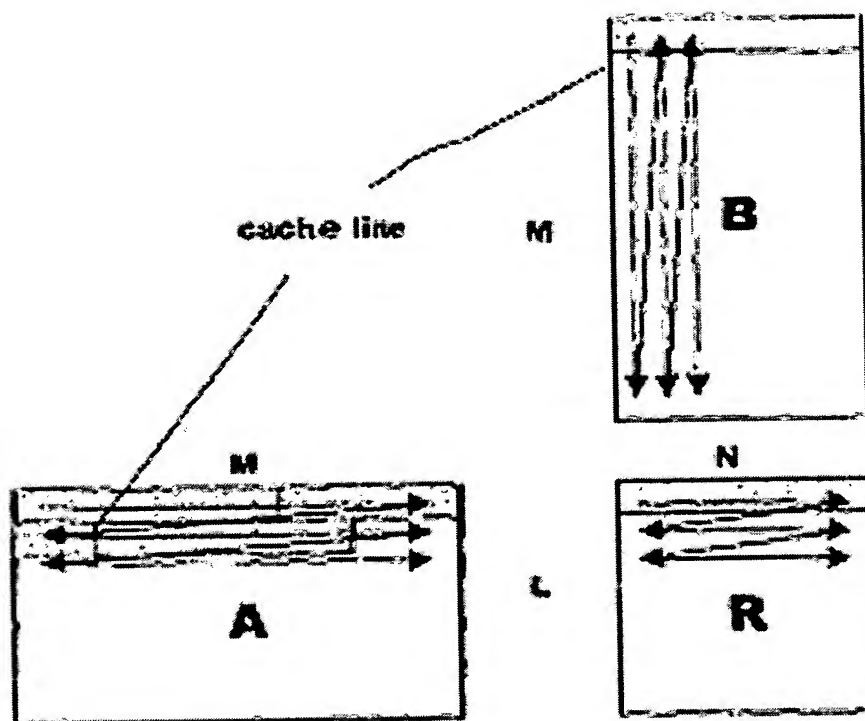


FIG. 35

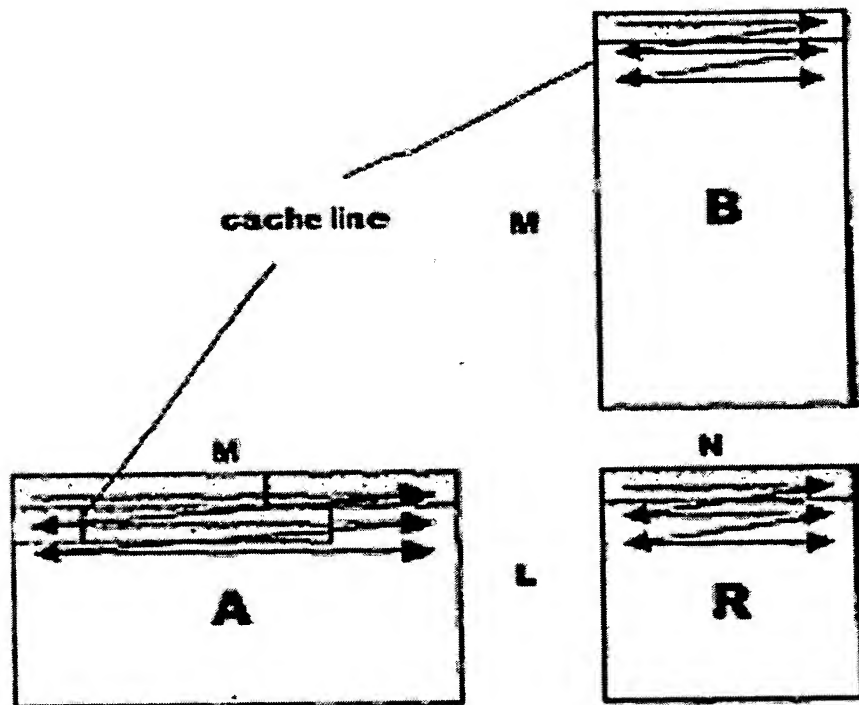


FIG. 36

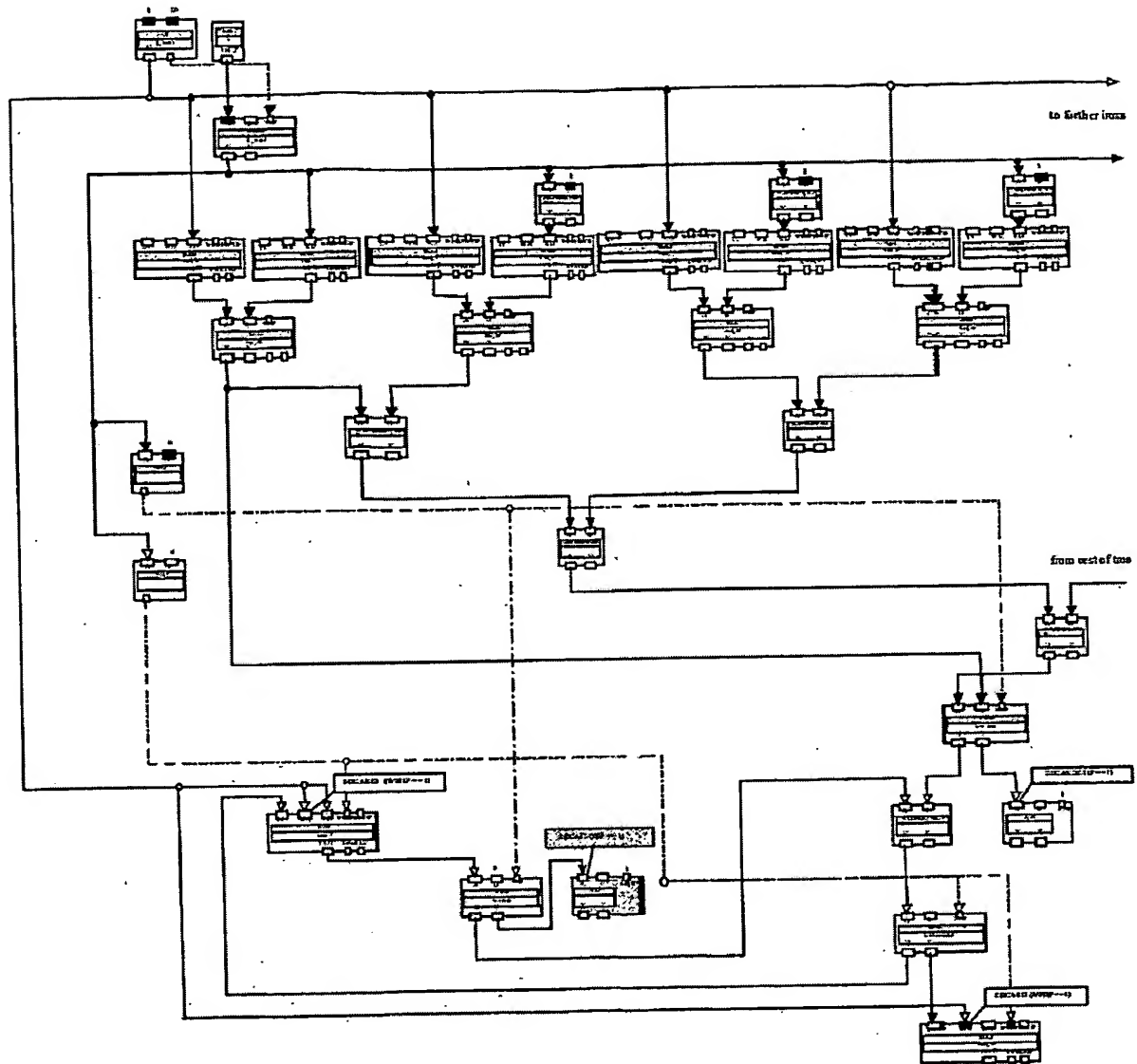


FIG. 37

NY01 1083323v1

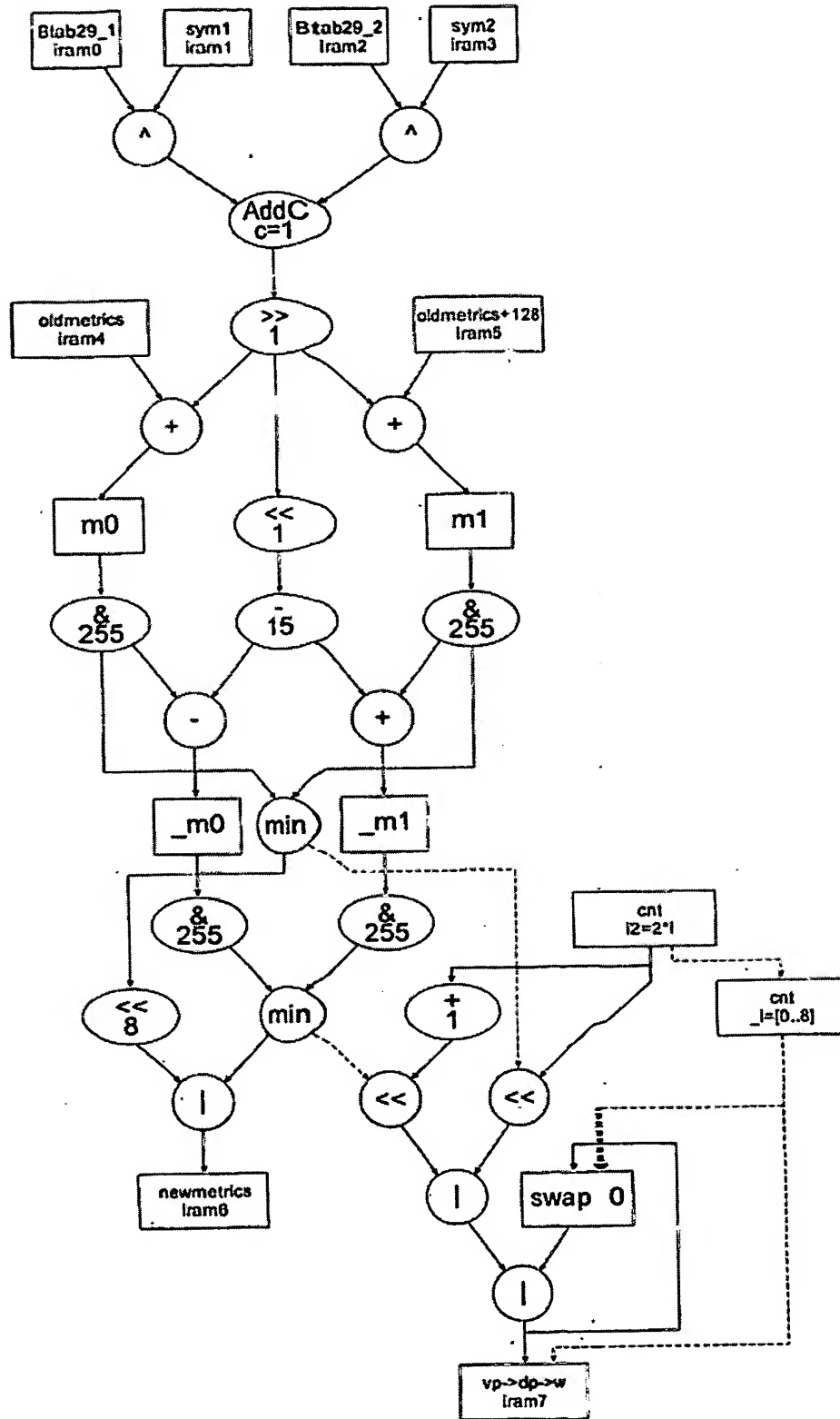


FIG. 39

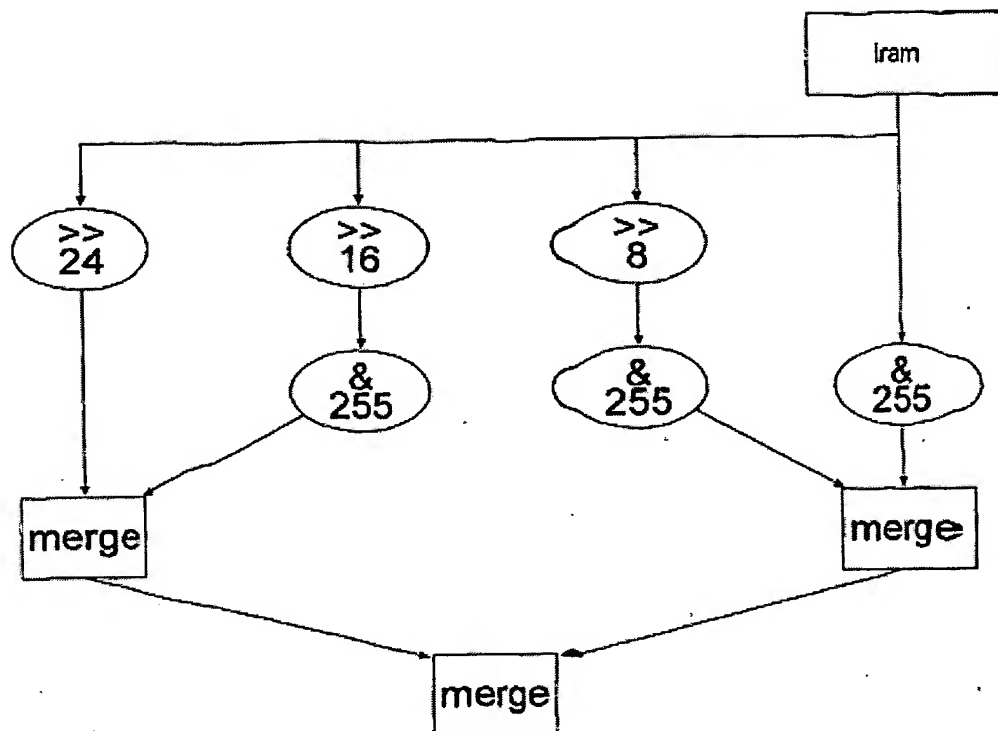


FIG. 40

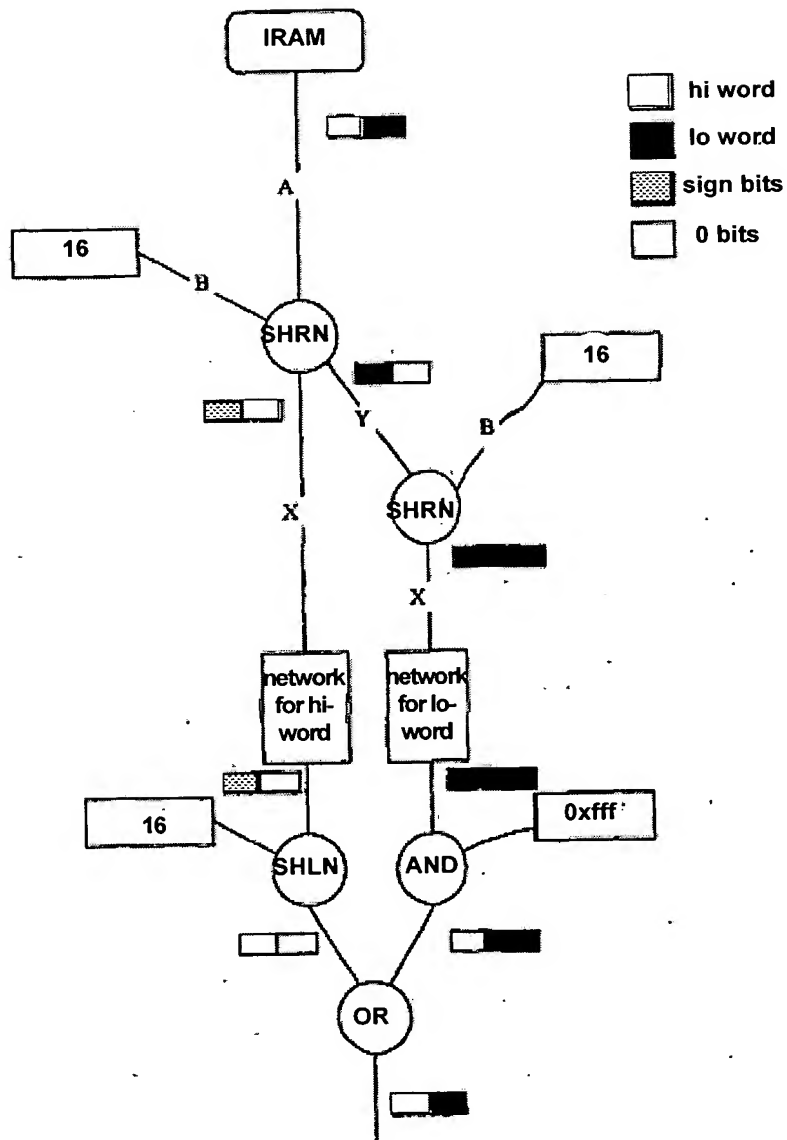


FIG. 41

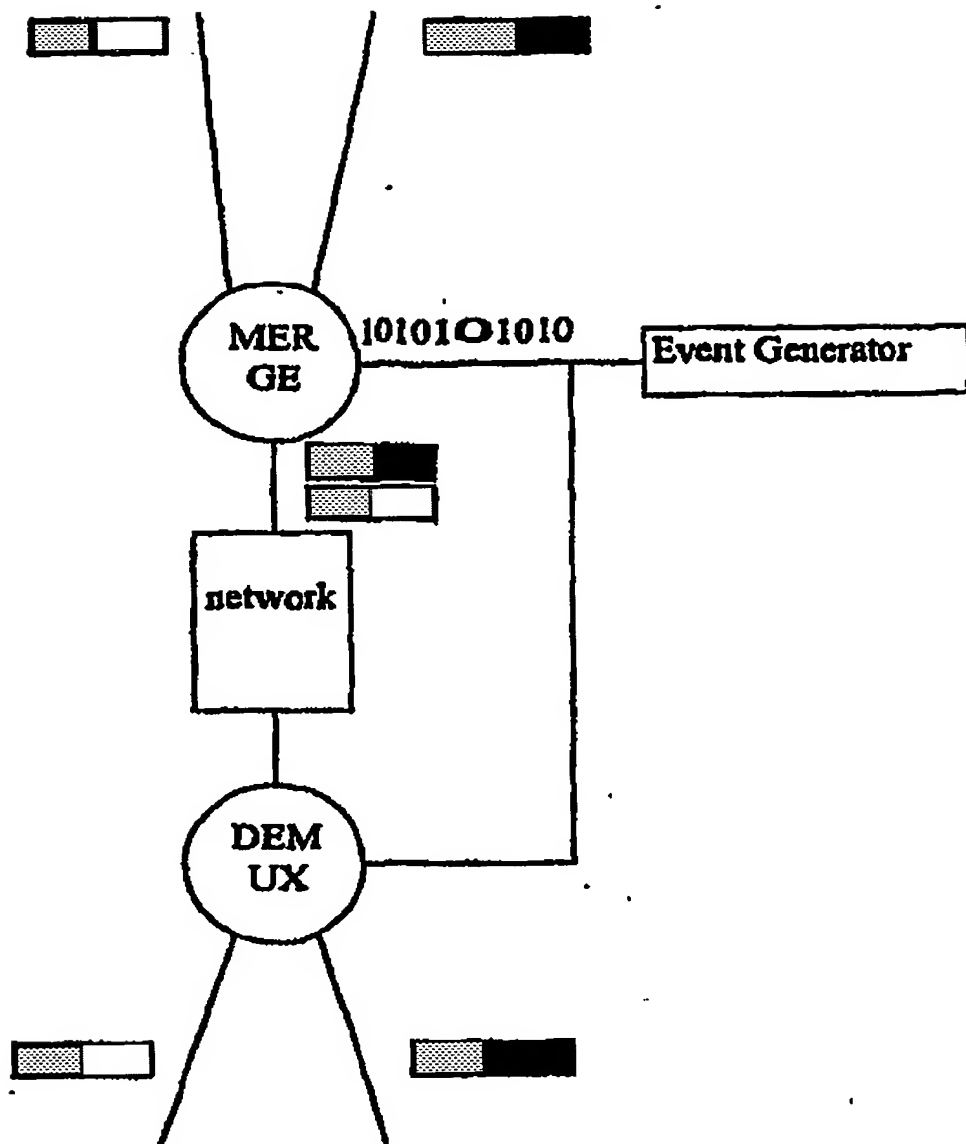


FIG. 42

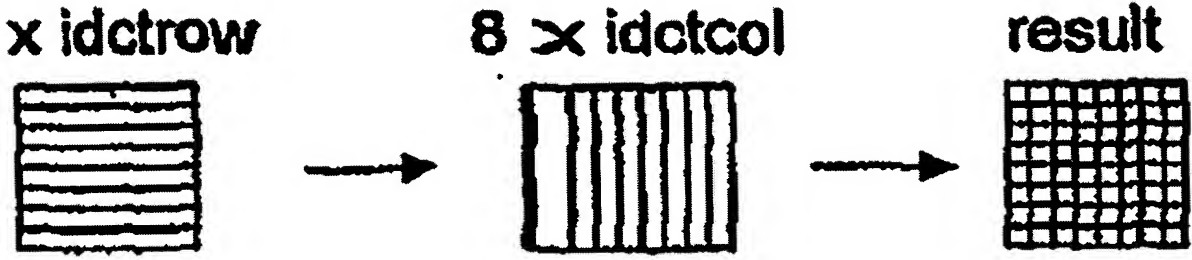


FIG. 43

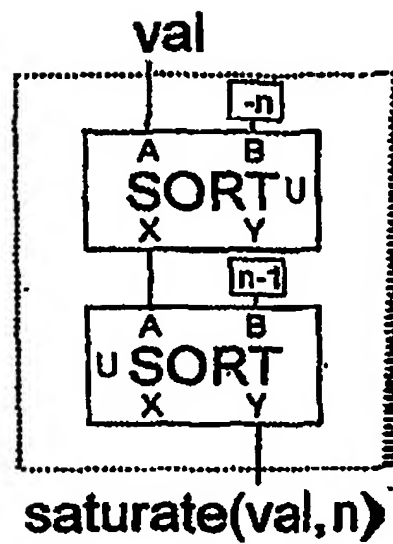


FIG. 44

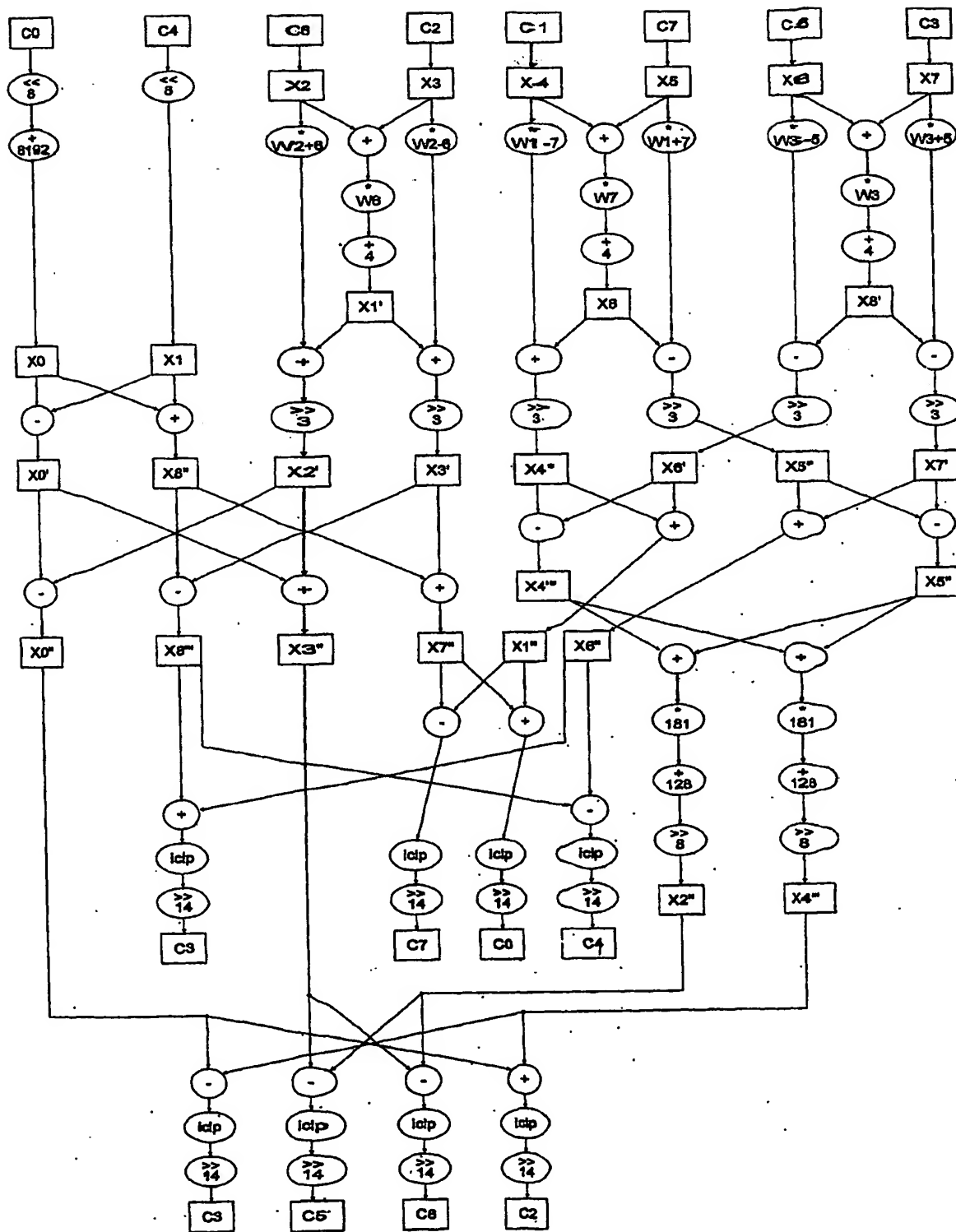


FIG. 45

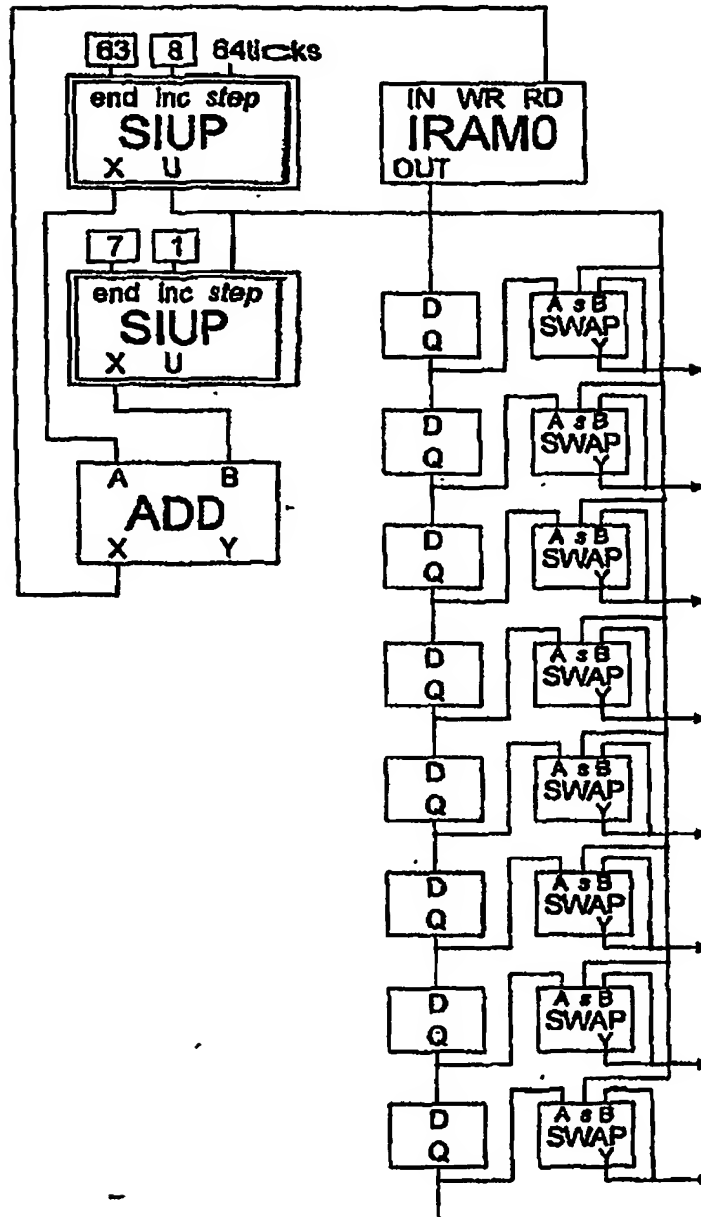


FIG. 46

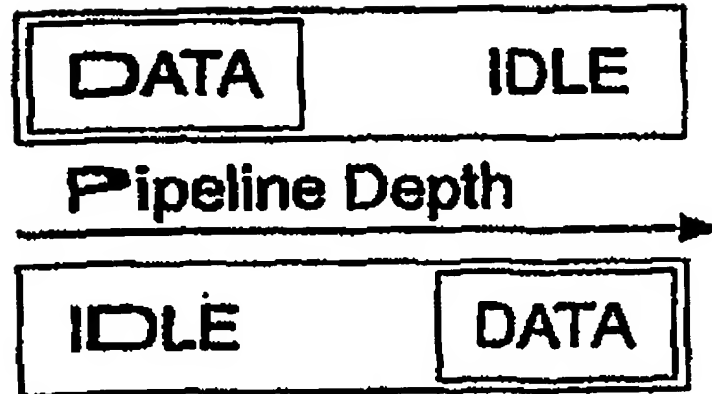


FIG. 47

```
// transform.c
..
for (n=0; n<block_count; n++) {
    idct(blocks[k*block_count+n]); // block_count is 6 or 8 or 12
}
..
// idct.c
/* two dimensional inverse discrete cosine transform */
void idct(block)
short *block;
{
    int i;

    for (i=0; i<8; i++)
        idctrow(block+8*i);

    for (i=0; i<8; i++)
        idctcol(block+i);
}
```

loop interchange

FIG. 48

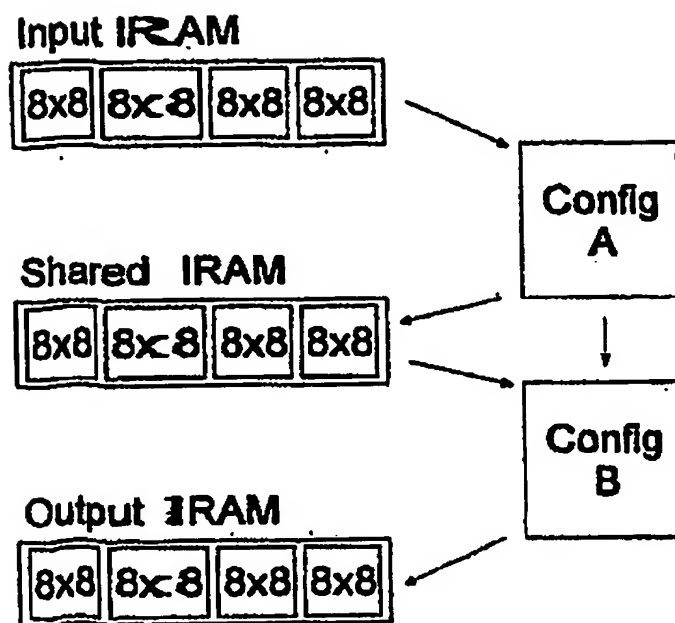


FIG. 49

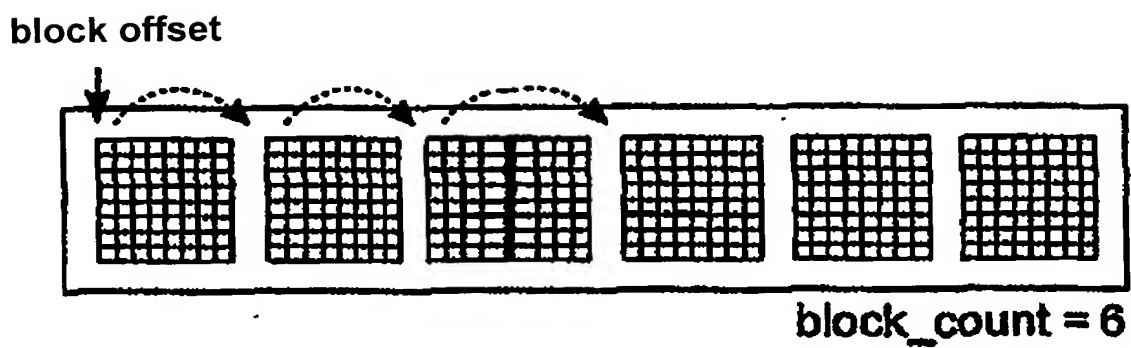


FIG. 50

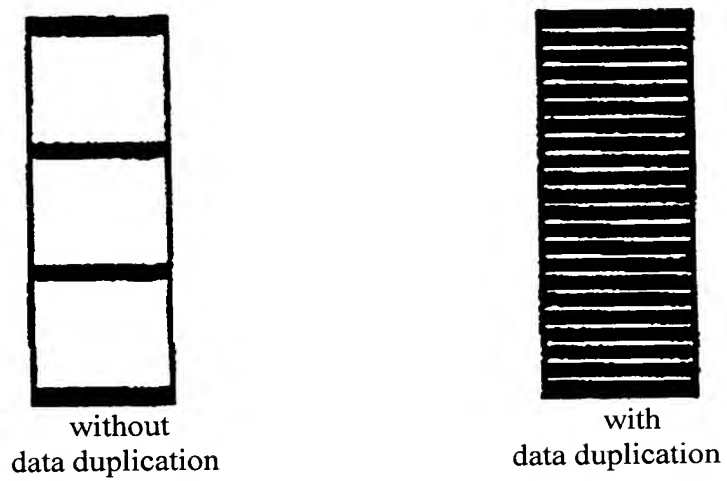


FIG. 51

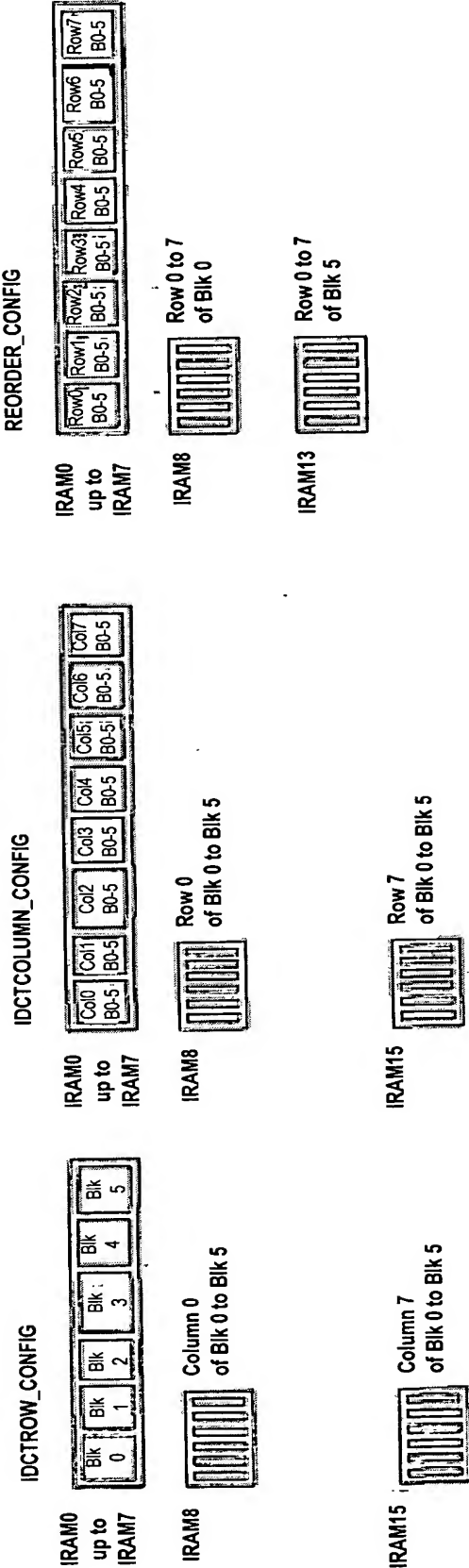


FIG. 52

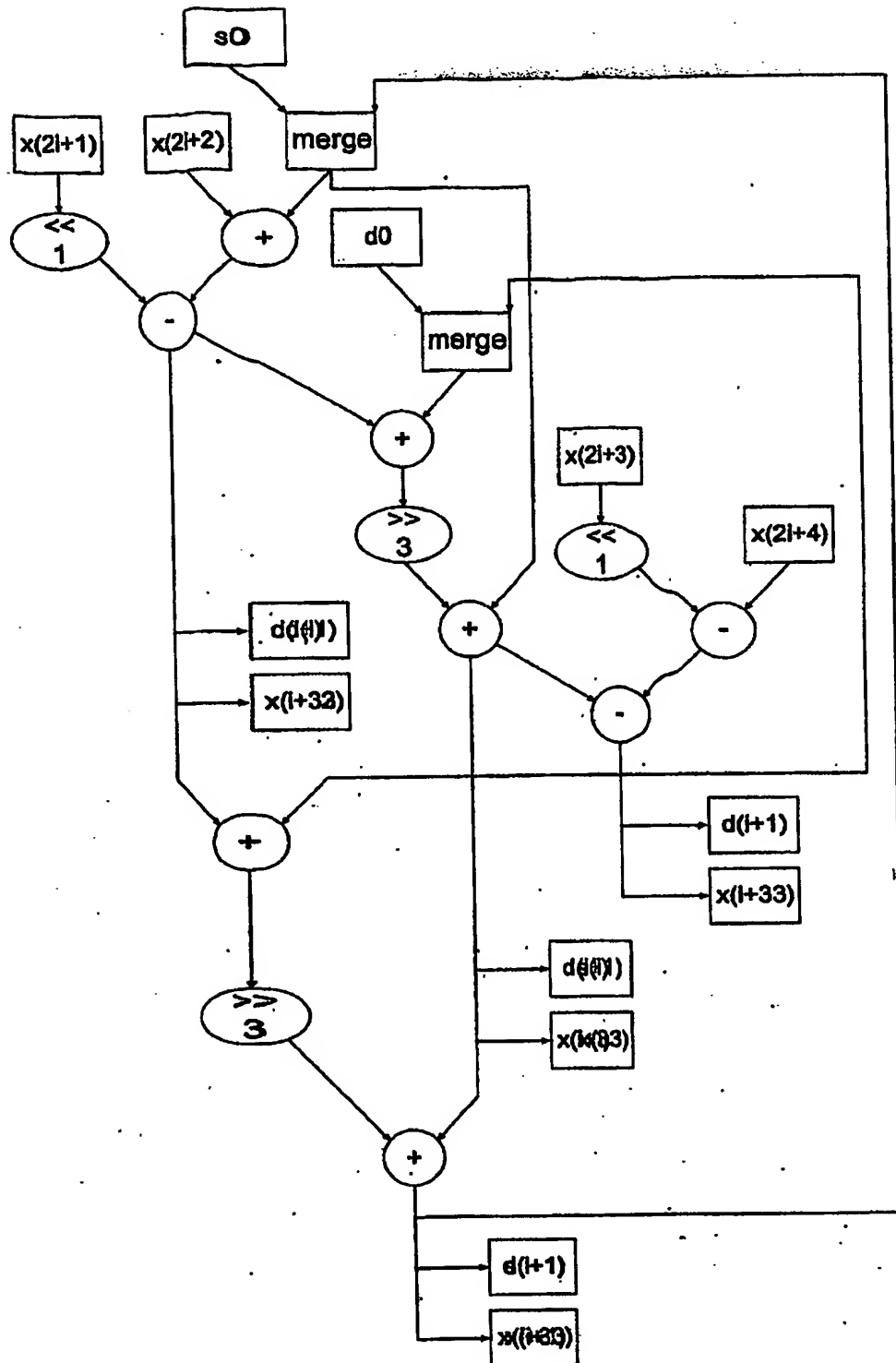


FIG. 53

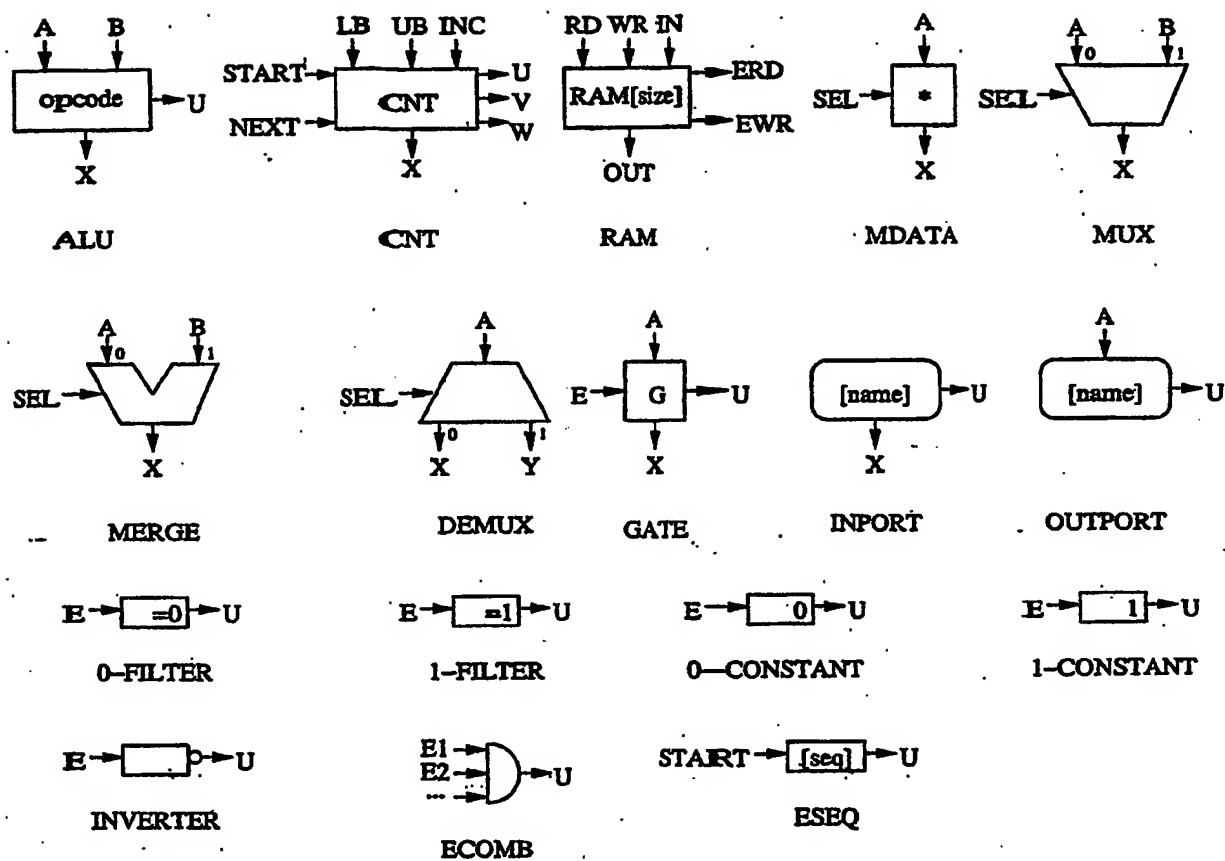


FIG. 54

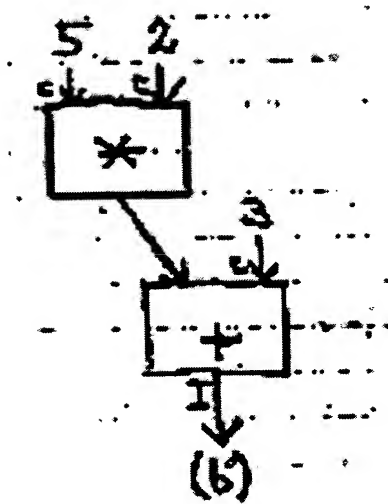


FIG. 55

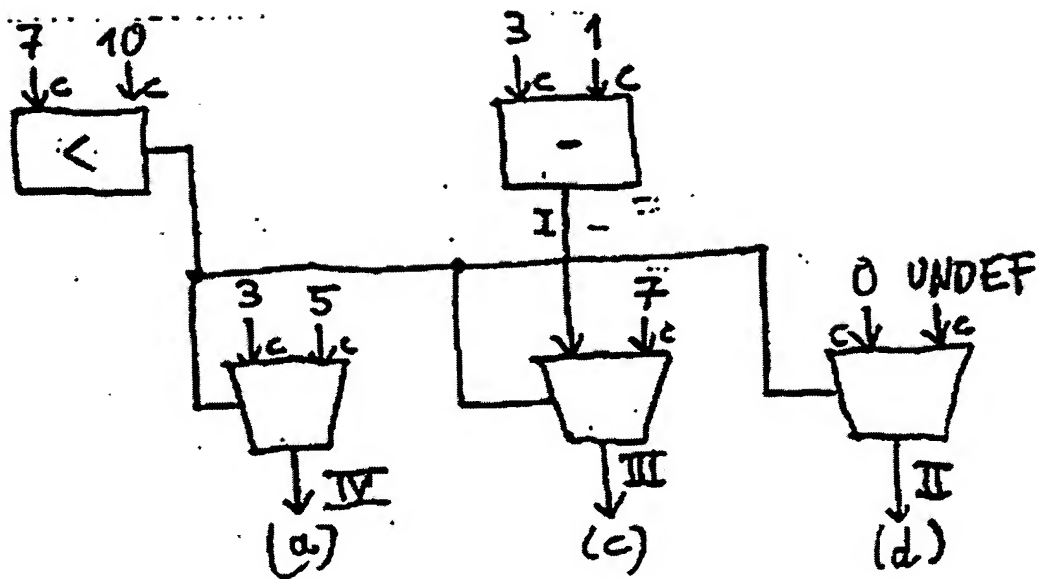


FIG. 56

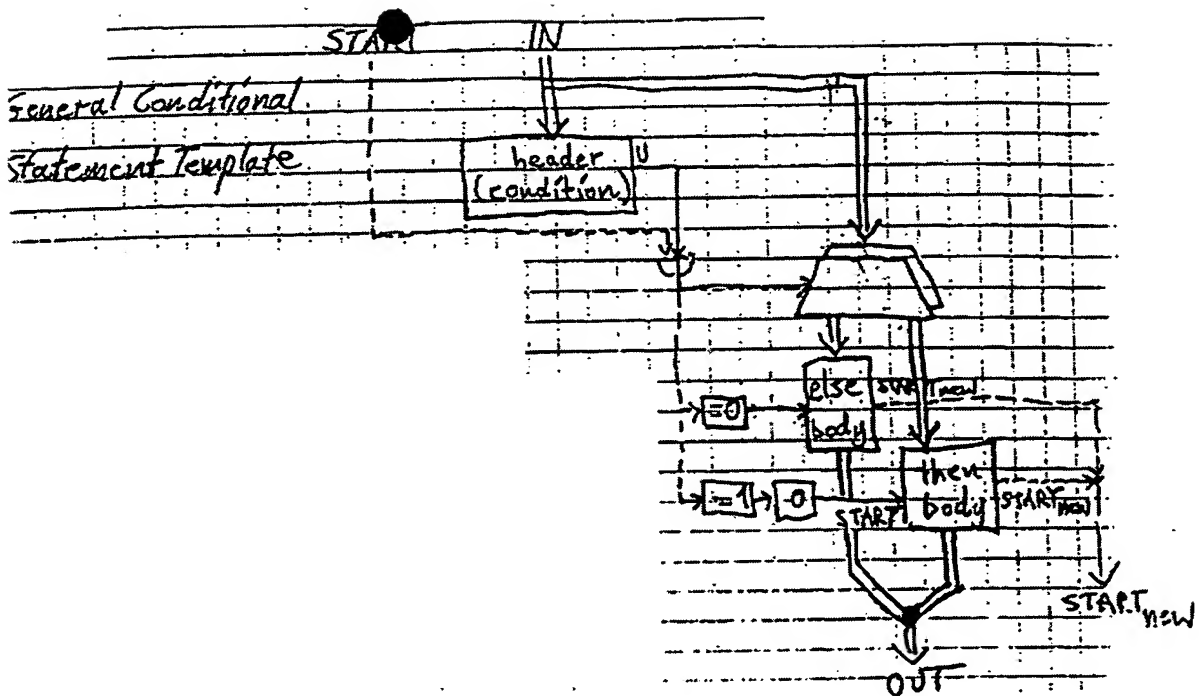


FIG. 57

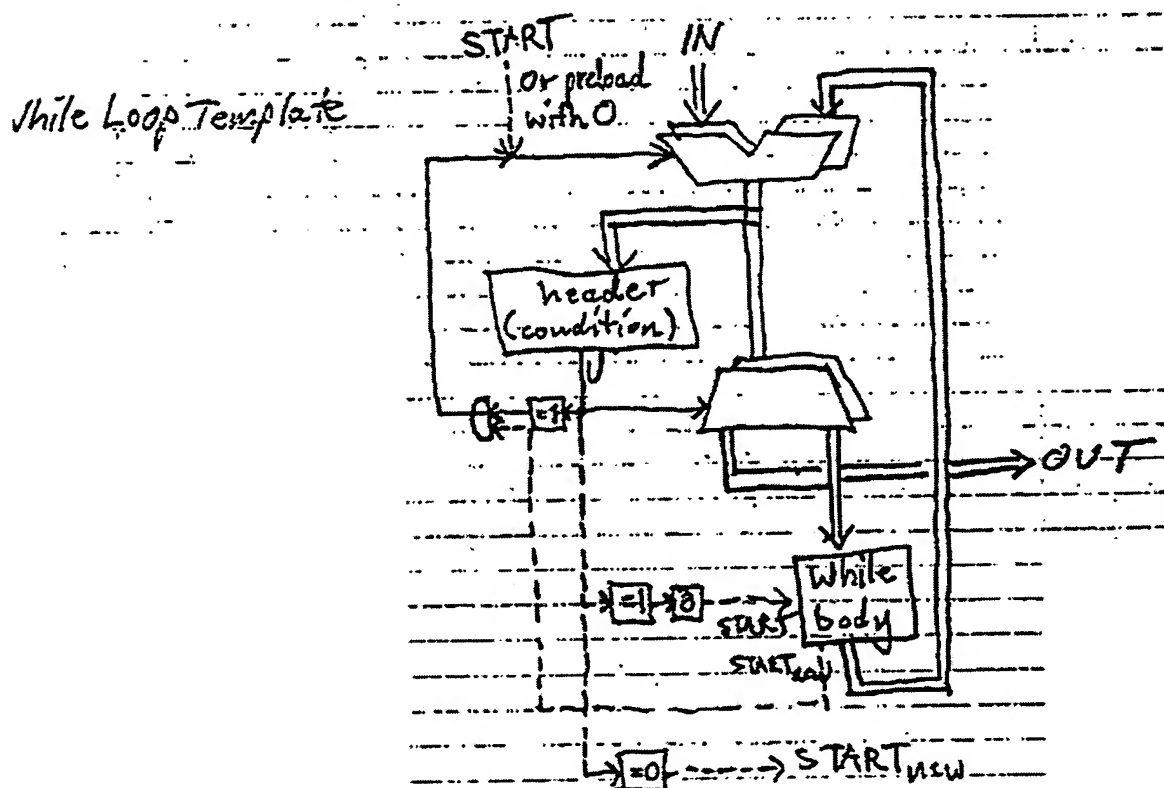


FIG. 58

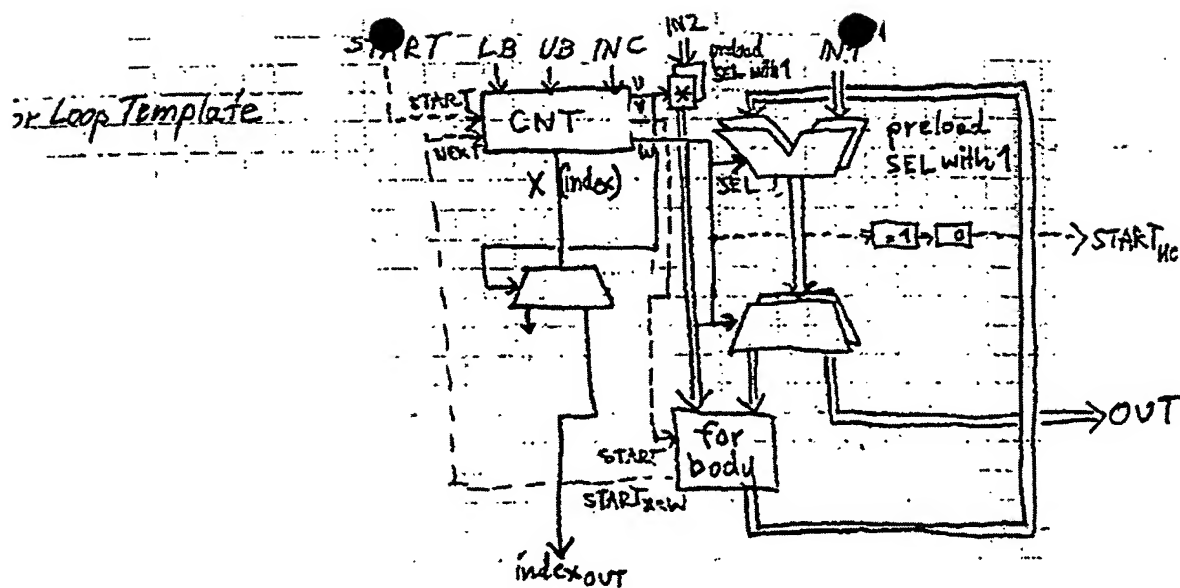


FIG. 59

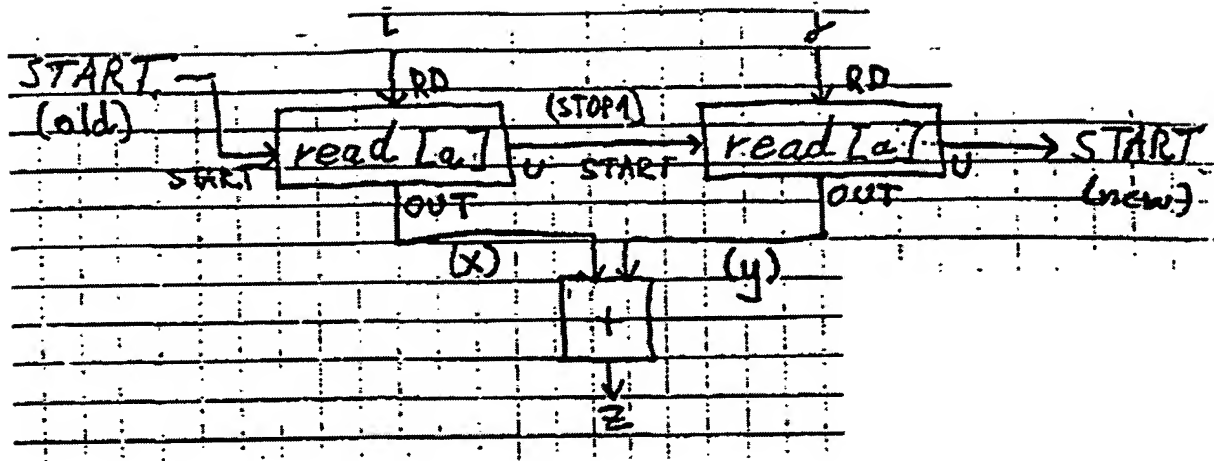


FIG. 60

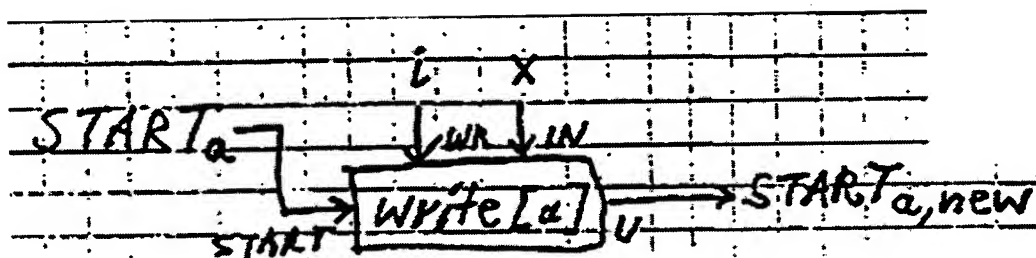


FIG. 61

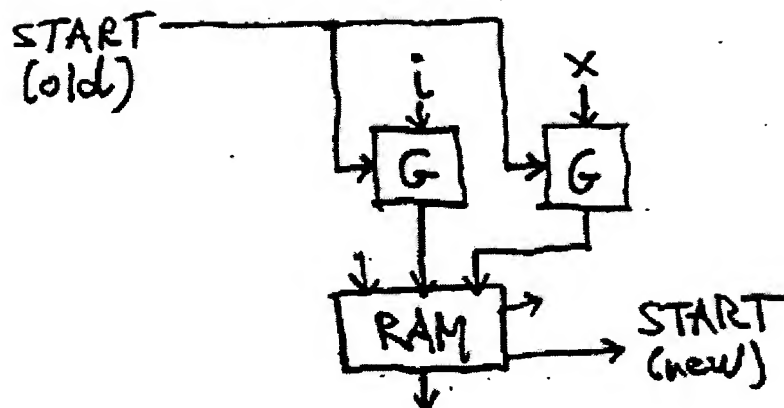


FIG. 62

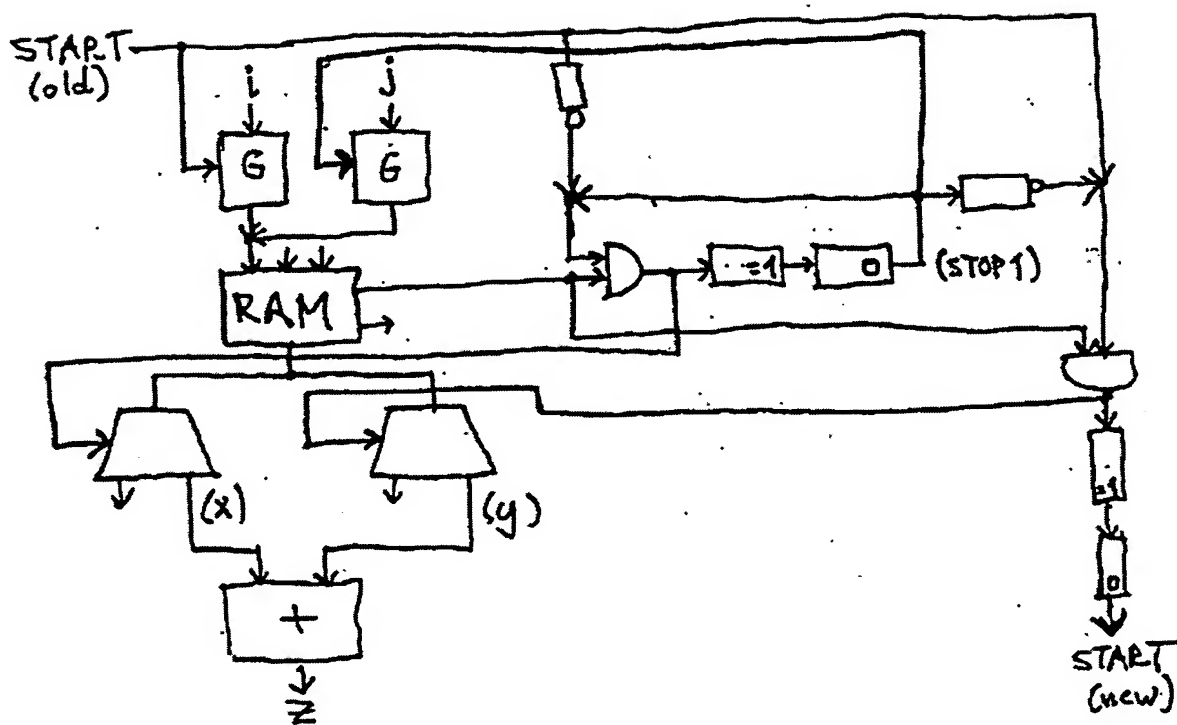


FIG. 63

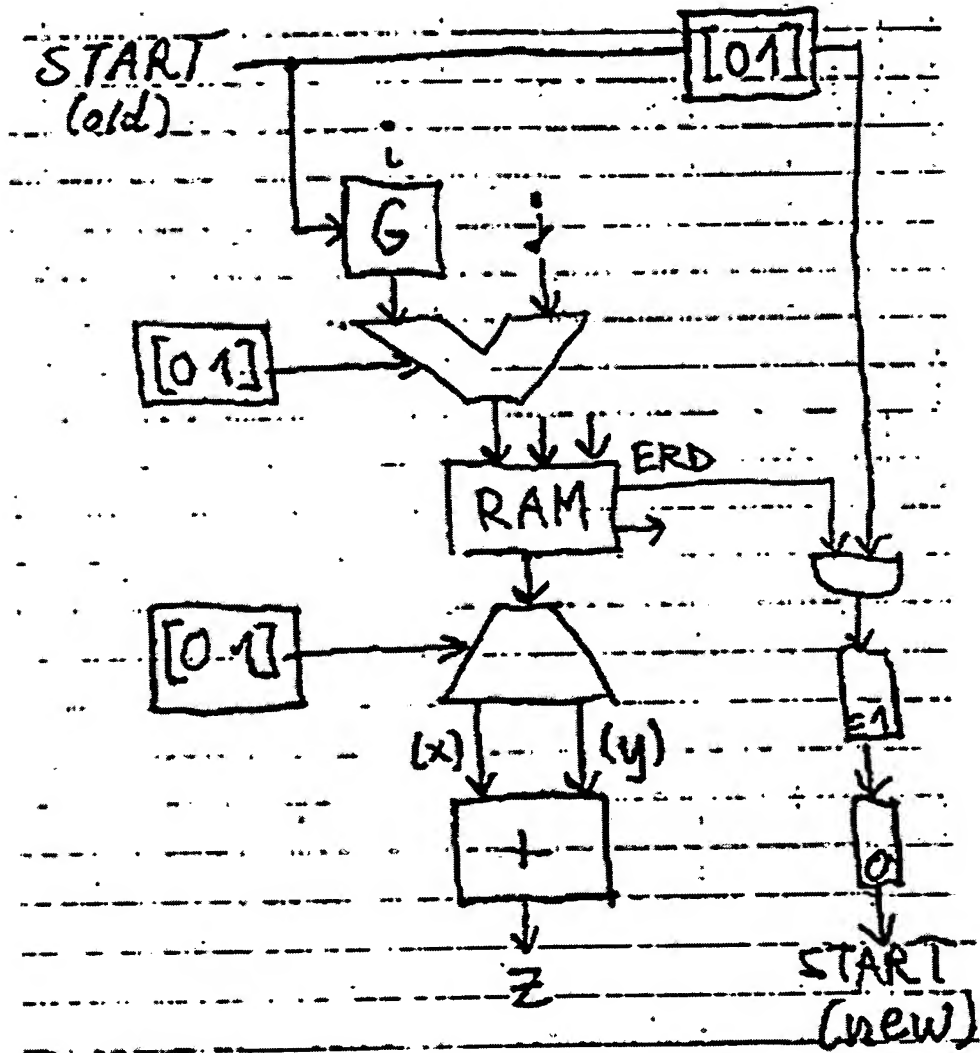


FIG. 64

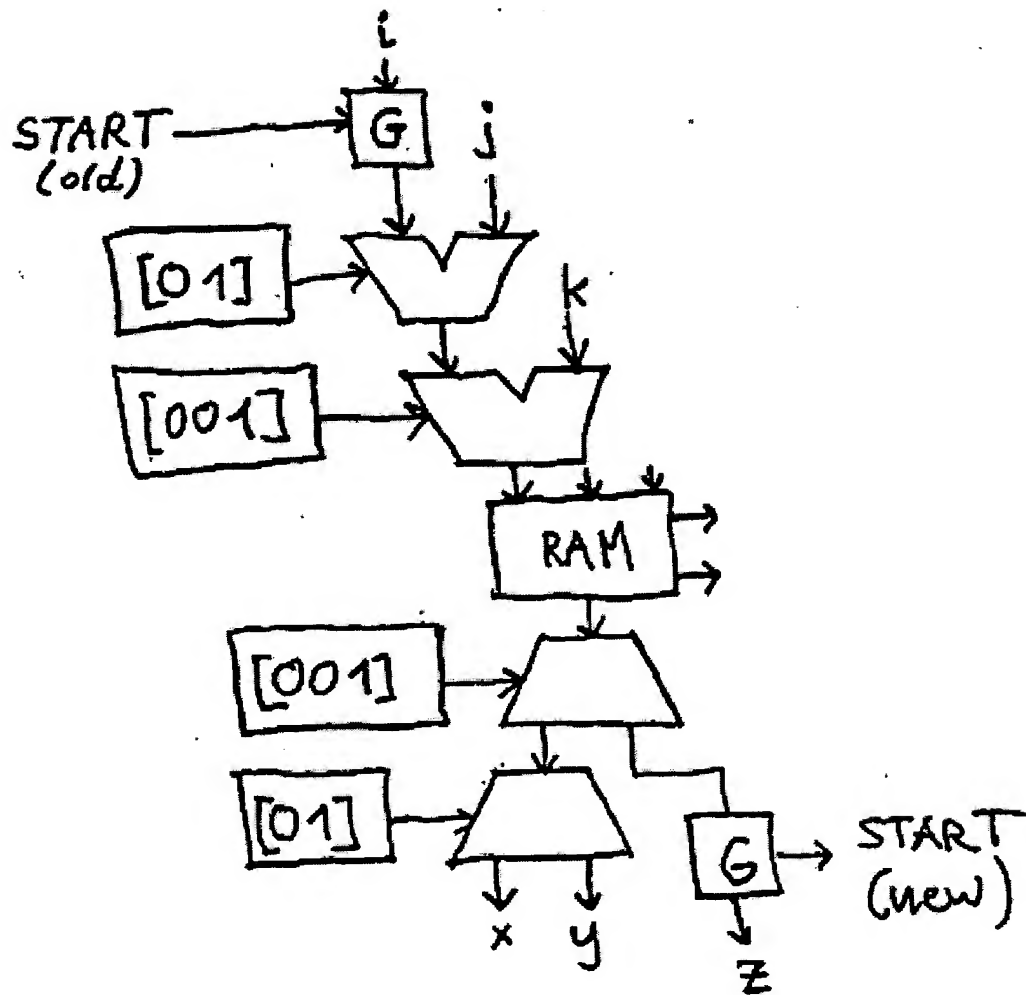


FIG. 65

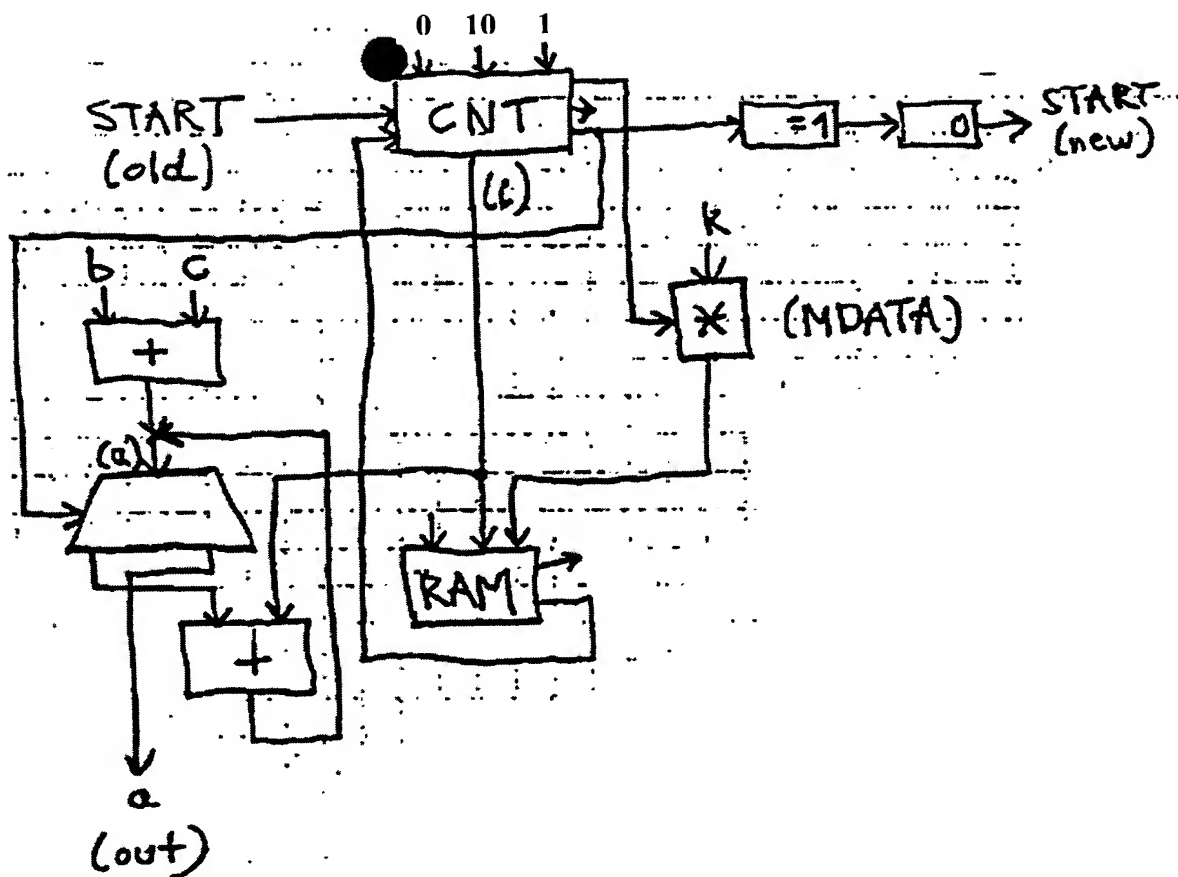


FIG. 66

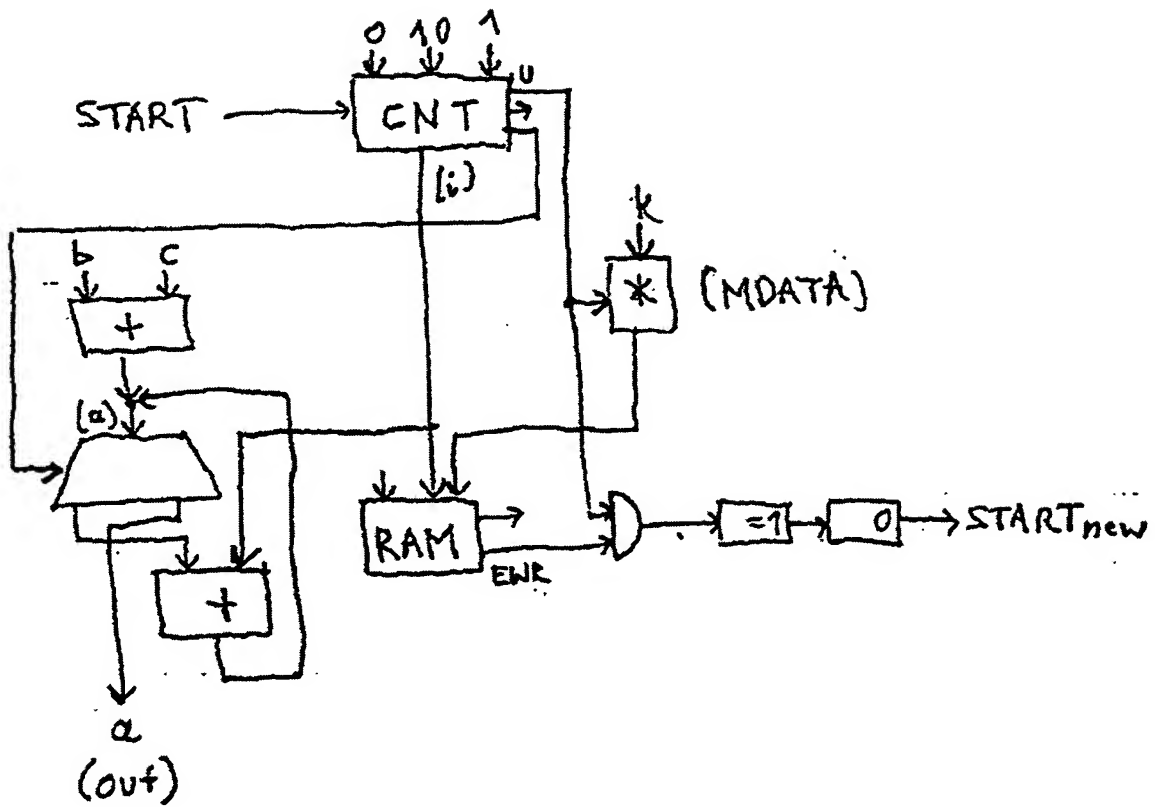


FIG. 67

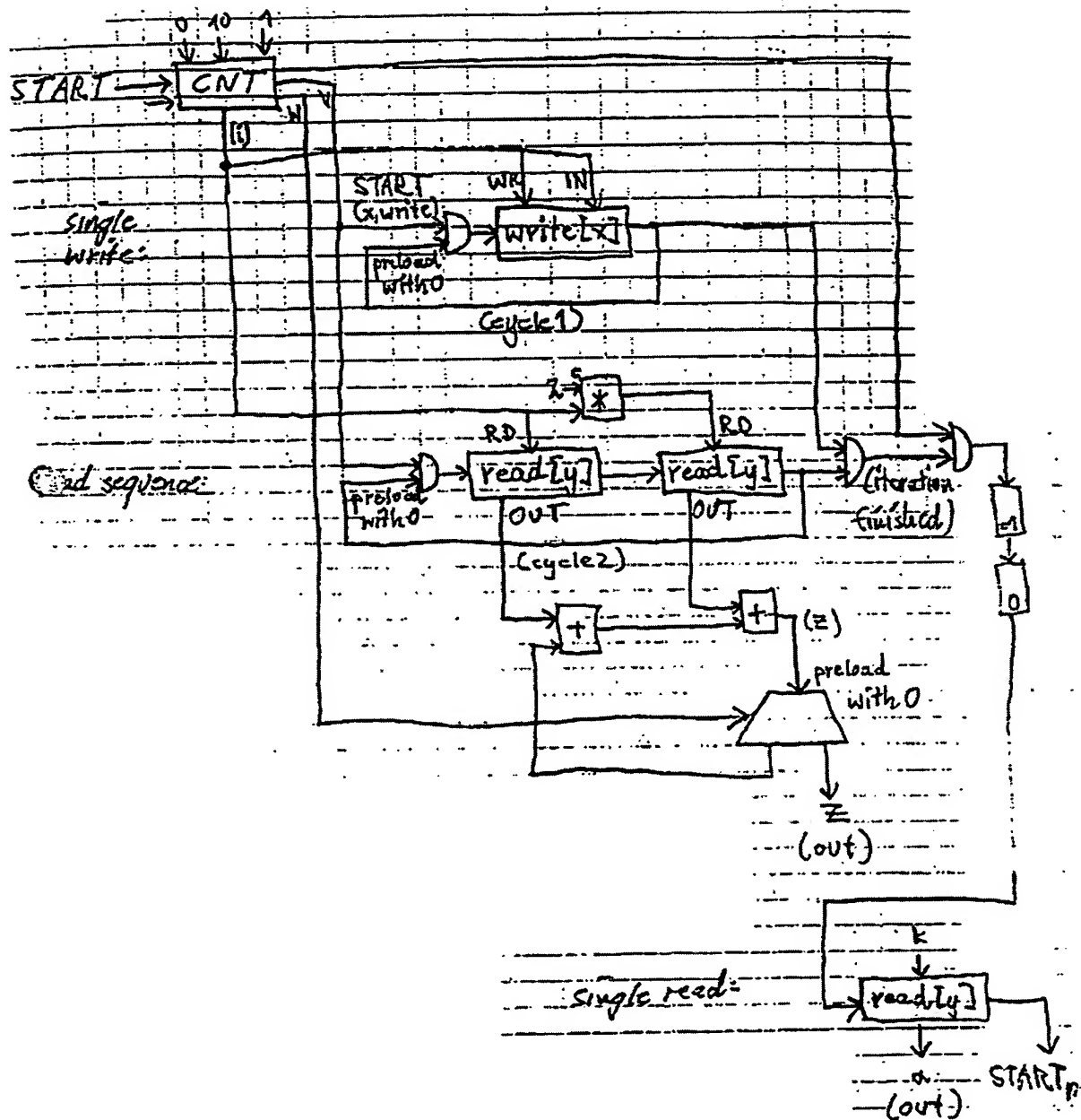


FIG. 68

NY01 1083323v1

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.